

# Editoriale

## I 50 anni della legge di Moore

L'integrazione circuitale costituisce una grande rivoluzione tecnologica. In precedenza, infatti, i circuiti elettronici erano realizzati con singoli componenti, fabbricati uno ad uno, e collegati poi tra loro. Con l'avvento del transistor e della tecnologia dei semiconduttori si apriva una straordinaria possibilità, ossia realizzare contemporaneamente tutti i componenti di un circuito, nonché i loro collegamenti, su un unico substrato. Nasceva cioè il concetto di circuito integrato.

Uno dei luoghi dove si sperimentavano queste nuove idee era la California, in quella che sarebbe poi stata chiamata la Silicon Valley. Lì lavorava Gordon Moore, un pioniere del transistor, che ebbe modo di incontrare durante un viaggio di lavoro agli inizi degli anni '60.

A quell'epoca i circuiti integrati erano ancora in fase embrionale. Per dare un'idea, il primo circuito integrato in silicio fu annunciato dalla Fairchild (di cui Moore era cofondatore) nel 1961 e conteneva soltanto quattro transistor.

Nel 1965 Moore scrisse un articolo su Electronics Magazine in cui faceva una previsione sui progressi della integrazione circuitale nel decennio a venire. Il suo approccio era molto concreto perché non considerava solo i possibili sviluppi della tecnologia, ma teneva conto anche degli aspetti economici e di mercato.

In fig.1 è riprodotto un grafico dell'articolo che mostra l'andamento del costo di un singolo transistor (o elemento circuitale) in funzione della complessità del chip. Come si vede, la curva ha un minimo. Infatti il costo dell'elemento diminuisce man mano che aumenta il grado di integrazione, ma ciò fino ad un certo punto, perché poi la resa del processo diminuisce e quindi il costo sale.

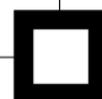
0

1

0

1

0



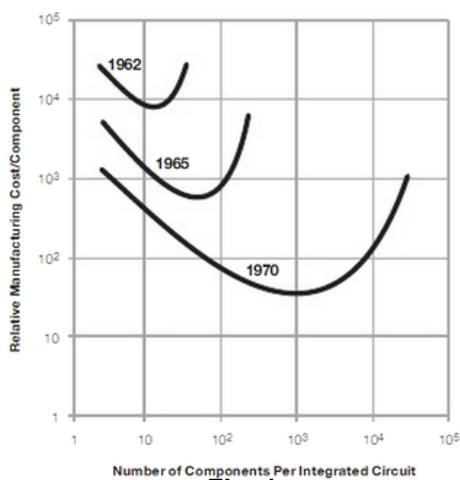


Fig. 1

Grafico dell'articolo di Moore che è alla base della sua "legge". Il numero di transistor da lui previsto per ogni anno corrisponde al punto di minimo costo. (G.E. Moore, "Cramming more components onto integrated circuits".

*Electronics Magazine*. 38(8), 1965, pp. 4-7).

Per ogni dato anno, il livello di integrazione previsto da Moore era quello per cui risultava minimo il costo del singolo transistor. Sulla base di questo rationale economico, nonché ovviamente delle proiezioni di sviluppo della tecnologia, Moore prevede che il numero di transistor per chip sarebbe raddoppiato ogni anno nell'arco del successivo decennio.

Nel 1975, ossia dieci anni dopo, la previsione risultava del tutto azzeccata. Quell'anno, Moore presentò ad un convegno un aggiornamento delle sue valutazioni. In sostanza, egli riteneva che il trend sarebbe continuato, seppure con un graduale rallentamento.

Alla luce dei fatti, anche questa previsione si è rivelata sostanzialmente

corretta. Il tempo di raddoppio della complessità dei circuiti integrati si è allungato dapprima a un anno e mezzo, per passare poi a circa due anni.

Questo trend si è verificato in particolare per i microprocessori, il cui primo esempio (che vale la pena di ricordare, porta la firma di Federico Faggin), fu prodotto nel 1971 e conteneva 2.300 transistor.

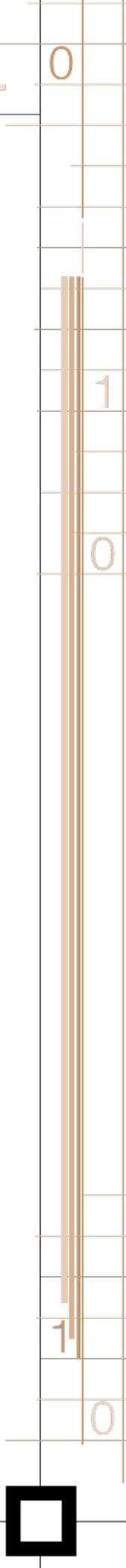
Oggi, sostanzialmente in accordo coi ritmi di crescita della complessità prima indicati, si realizzano industrialmente su un chip oltre 10 miliardi di transistor.

Un progresso davvero epocale quello della microelettronica, che ha aperto possibilità straordinarie che sono sotto gli occhi di tutti.

Va detto che la proiezione fatta da Moore, ormai 50 anni fa, non è una legge di natura. In effetti, sulla evoluzione della complessità dei chip intervengono non solo fattori di natura tecnologica, ma anche altri, come le strategie dell'industria del settore. Comunque, il riscontro con quanto effettivamente avvenuto può giustificare l'appellativo di "legge" attribuito alle previsioni di Moore.

Per quanto tempo ancora varrà la legge di Moore? E' difficile dare una risposta precisa; per mantenere il ritmo intervengono molti fattori: ricerca sui materiali, nuovi processi produttivi, strumentazioni sempre più sofisticate, investimenti crescenti. Si ritiene comunque che ci sia ancora spazio prima che limiti fisici ed economici ne decretino il tramonto. Un tramonto più volte annunciato in passato, ma finora smentito dai fatti. Lunga vita alla legge di Moore!

Franco Filippazzi  
AICA



# Programmazione parallela: evoluzione e nuove sfide

Marco Danelutto

## Sommario

*Le recenti evoluzioni dei componenti di calcolo hanno messo a disposizione processori sempre più veloci e potenti che pongono nuove sfide dal punto di vista degli strumenti di programmazione. I classici strumenti utilizzati fino ad ora, infatti, richiedono un notevole sforzo di programmazione quando si vogliono sfruttare appieno tutte le caratteristiche dei nuovi componenti. In questo lavoro discutiamo alcuni recenti sviluppi nei modelli di programmazione parallela strutturata che permettono di sviluppare rapidamente applicazioni molto efficienti in grado di utilizzare tutte le nuove caratteristiche disponibili.*

## Abstract

*Recent advances in digital components provided faster and more powerful computing devices that open new challenges to the programming tools. Classical programming tools require quite a programming effort in all those cases where the advanced features of these new components must be exploited. In this work, we discuss some recent advances in the field of the structured parallel programming models that support the rapid development of efficient parallel applications suitable to exploit all the new features available.*

**Keywords:** Parallel programming, structured parallel programming models, algorithmic skeletons, design patterns, multi-core, GPU, FPGA.



## 1. Introduzione

### 1.1 Dal sequenziale al parallelo

Fino ai primi anni 2000 l'evoluzione dell'architettura dei componenti di calcolo ha messo a disposizione processori sempre più potenti in grado di eseguire sempre più velocemente gli stessi programmi utilizzati sui processori delle generazioni precedenti. Questo succedeva perché il *modello architetturale* del processore, ovvero l'insieme di caratteristiche che ne determinano le funzionalità, non cambiava mai. Si trattava di un modello nel quale un insieme di componenti hardware era in grado di eseguire istruzioni di un unico programma alla volta. Questo modello architetturale era caratterizzato dall'esistenza di un unico dispositivo in grado di eseguire in continuazione il cosiddetto ciclo *fetch-decode-execute*, ovvero di prelievo dalla memoria dell'istruzione all'indirizzo indicato da un registro detto program counter (fetch), di decodifica dell'istruzione per capirne il tipo (decode) e infine di esecuzione dell'istruzione stessa, con il conseguente aggiornamento del program counter (execute).

Questo semplice modello architetturale è evoluto nel tempo introducendo forme di parallelismo (esecuzione contemporanea e concorrente di diversi passi di calcolo) nell'esecuzione delle singole istruzioni che compongono un programma:

- si è fatto in modo che l'esecuzione degli stadi del ciclo fetch-decode-execute fossero eseguite da unità indipendenti, in modo da sovrapporre l'esecuzione di un'istruzione con la decodifica dell'istruzione successiva e con il prelievo dalla memoria dell'istruzione ancora successiva, secondo un modello simile a quello della catena di montaggio, detto *pipeline*. Successivamente si sono suddivisi ulteriormente tutti gli stadi del pipeline fino ad arrivare a pipeline di esecuzione con decine di stadi, tutti in grado di lavorare contemporaneamente su fasi diverse di diverse istruzioni dello stesso programma;
- si sono replicate le risorse interne del processore in modo da poter eseguire più istruzioni dello stesso programma contemporaneamente (modello di architettura *superscalare*);
- si sono introdotte migliorie nel sottosistema di memoria (programma e dati) che hanno portato ad una riduzione notevole del tempo necessario per l'accesso alla singola istruzione o ai dati in memoria (introduzione di uno o più livelli di cache).

Sebbene vi siano limitazioni dovute alle dipendenze fra istruzioni diverse dello stesso programma che limitano l'efficacia complessiva sia del modello pipeline che del modello superscalare, in linea di principio l'adozione di un modello pipeline a  $k$  stadi porta ad uno *speedup* (incremento della velocità di esecuzione) pari a  $k$ , così come l'adozione di un modello superscalare a  $k$  vie (capace cioè di eseguire  $k$  istruzioni alla volta) porta ugualmente ad uno *speedup* pari a  $k$ . Le due tecniche possono essere combinate fra loro.

Le migliorie nel modello architetturale di cui stiamo parlando sono state rese possibili dalla sempre maggiore densità di componenti elettronici sul silicio. I

componenti di calcolo, infatti, sono realizzati “stampando” con veri e propri processi litografici le porte logiche con cui sono realizzati i componenti direttamente sulla superficie di un cristallo di silicio. La legge di Moore, formulata negli anni '60 da Gordon Moore, co-fondatore di Intel, e rispettata fino ai giorni nostri, stabilisce che il numero di componenti (porte logiche) su una certa superficie del wafer di silicio raddoppi ogni due anni, per via dei miglioramenti delle tecnologie di produzione [1,2]. Questo ha permesso di realizzare sempre nuovi modelli architetturali che occupano sempre maggiori aree sul chip, come per esempio i modelli pipeline/superscalare appena menzionati. Ma i miglioramenti nella tecnologia di produzione dei chip hanno portato anche ad una maggior miniaturizzazione dei componenti e dunque alla possibilità di lavorare a frequenze di clock sempre maggiori. Dal momento che il clock determina in modo direttamente proporzionale la velocità finale di esecuzione delle istruzioni, questo ha portato a dispositivi che non solo erano in grado di eseguire più velocemente istruzioni per via dei miglioramenti del modello architetturale, ma le eseguivano sempre più in fretta a parità di modello per via delle frequenze dei clock sempre più alte.

Questo trend ha subito un brusco rallentamento nei primi anni 2000 per due ragioni fondamentali:

- non si è stati più in grado di trovare miglioramenti del modello architetturale del processore che permettessero incrementi sostanziali nelle prestazioni dell'esecuzione delle istruzioni provenienti da un unico programma, o perlomeno non si è stati in grado di trovare miglioramenti architetturali che portassero ad un incremento delle prestazioni proporzionale all'area di silicio occupata per introdurre tali miglioramenti. In altre parole, l'introduzione di nuove caratteristiche che potevano portare ad un incremento delle prestazioni di un X% a fronte di un corrispondente incremento dell'area di silicio necessaria del 10X% [3];
- l'utilizzo di frequenze di clock sempre più alte per migliorare le prestazioni delle CPU (Central Processing Unit) ha introdotto notevoli problemi di consumo e conseguentemente di dissipazione termica. I nuovi chip con clock che viaggiavano ormai speditamente verso i 5-6Ghz richiedevano tecnologie di raffreddamento complicate e costose, sia in termini di costruzione che di esercizio (ulteriori Watt richiesti non per far funzionare il processore ma per dissipare il calore prodotto).

Dunque a cavallo fra il secolo scorso e quello attuale i produttori si sono trovati di fronte a una situazione che vedeva da una parte la possibilità di realizzare componenti di calcolo sempre più complessi su un singolo chip e dall'altra parte la difficoltà oggettiva a realizzare nuovi componenti in grado di sfruttare queste potenzialità nella realizzazione di processori in grado di lavorare sempre più velocemente su singolo programma alla volta.

Il riconoscimento di questa situazione ha portato ad un cambio radicale di strategia:

---

non si è più cercato di sfruttare le maggiori capacità relative alla realizzazione di componenti hardware per realizzare processori in grado di eseguire più velocemente le istruzioni di un singolo programma, ma si sono sfruttate queste capacità per realizzare sullo stesso chip più core cioè più componenti che, operando in modo combinato ed accedendo ad un sottosistema di memoria comune, sono ognuno in grado di eseguire istruzioni di un programma diverso.

---

Questa svolta è radicale non perché cambia il tipo di componente, ma perché si passa da una situazione in cui un programmatore può concepire il suo codice come una serie di attività (istruzioni) da eseguire sequenzialmente, una dopo l'altra ad una situazione in cui il programmatore *deve* concepire il proprio programma come *un insieme di attività* in grado di essere eseguite in modo autonomo da core diversi, *coordinate esplicitamente* per far sì che la loro esecuzione porti al calcolo del risultato finale richiesto dall'applicazione.

All'inizio dell'era multi-core questo cambiamento non fu percepito in modo chiaro: i core disponibili, e dunque le attività necessarie per utilizzarli al meglio, furono dell'ordine delle unità e fu relativamente facile utilizzarli per eseguire in parallelo codice del sistema operativo e codice della/delle applicazioni utente. In questo caso, non vi era necessità di modificare né il codice del sistema operativo né quello delle applicazioni stesse. Subito dopo però, gli avanzamenti nella tecnologia di produzione, che continua a progredire secondo la legge di Moore "modificata" (si raddoppia il numero di core sullo stesso chip circa ogni anno), hanno permesso di realizzare decine e poi centinaia di core sullo stesso chip. Con simili numeri la necessità di programmi *paralleli*, ovvero costruiti in modo da implementare un certo numero di attività parallele da eseguire sui vari core a disposizione, è diventata assoluta ed urgente.

Ma vi è un ulteriore fattore che ha reso e rende tale necessità urgente. Per riuscire a realizzare più core sullo stesso chip, i produttori hanno cominciato ad eliminare dai singoli core tutte quelle caratteristiche che introducevano miglioramenti minimali nei tempi di esecuzione a fronte di un'occupazione significativa in termini di area sul silicio. Dunque prendendo un programma che raggiunge una certa velocità su un processore non multi-core ed eseguendolo su un singolo core dei nuovi processori multi-core si otterrebbero prestazioni (leggermente) inferiori. Questo interrompe di fatto il trend, in essere fino ai primi anni del secolo, che vedeva la possibilità di far girare il proprio codice sempre più velocemente semplicemente comprando l'ultimo modello di processore.

Sebbene questo tipo di effetto si possa verificare semplicemente fra generazioni diverse della stessa serie di processori (per esempio fra processori multi-core Intel della serie Nehalem e della successiva serie Sandy Bridge), l'effetto è particolarmente marcato nei cosiddetti *general purpose many core*. Questi dispositivi sono chip che raggruppano da decine fino a centinaia di core, di capacità più limitate rispetto ai core delle CPU tradizionali, interconnessi con reti

ad anello o a mesh in grado di eseguire ciascuno un flusso di controllo programmato con strumenti di programmazione tradizionali. Un esempio di questa classe di dispositivi è l'Intel® XEON PHI, dotato di 60 core, ciascuno 4-way hyperthreading (cioè in grado di supportare l'esecuzione contemporanea di quattro thread distinti) e dotato di unità vettoriale da 512 bit, interconnessi secondo una topologia ad anello e dotati di cache di secondo livello condivisa e coerente [4]. I singoli core del PHI sono in grado di eseguire codice x86, ma, sia per via della semplificazione del progetto architetturale del core legata alla razionalizzazione dello spazio sul chip, sia per via della frequenza di clock piuttosto bassa (intorno al GHz), l'esecuzione del codice sul singolo core risulta sensibilmente più lenta dell'esecuzione dello stesso codice su un core di una CPU tradizionale della stessa generazione. Questo non è altro che un ulteriore fattore che spinge nella direzione della ricerca di mezzi di programmazione parallela sempre più efficienti nell'utilizzo concorrente di tutte le risorse a disposizione sul dispositivo.

## 1.2 Dall'omogeneo all'eterogeneo

Il passaggio dal supporto all'esecuzione di un singolo flusso di controllo a quello di flussi di controllo multipli non è l'unico cambiamento significativo dettato dagli avanzamenti della tecnologia di produzione dei componenti. Vi sono almeno altri due cambiamenti che hanno avuto un grosso impatto sullo scenario attuale:

- la disponibilità di GPU (Graphics Processing Units), ovvero di acceleratori in grado di eseguire molto velocemente tutte le operazioni tipiche della grafica computazionale. Le GPU utilizzano un gran numero di componenti (migliaia su una singola GPU) a loro volta capaci di eseguire ciascuno semplici operazioni sul singolo pixel di un'immagine [5]. Vista la particolare efficacia delle GPU nel calcolo di quelle computazioni che richiedono di eseguire una singola funzione su tutti gli elementi di una matrice o di un vettore, le GPU si sono presto evolute in GP-GPU (General Purpose GPU) in grado di accelerare notevolmente l'esecuzione di calcoli *data parallel*, cioè calcoli in cui il risultato della computazione si ottiene combinando i risultati da sotto computazioni effettuate su partizioni del dato in input;
- la disponibilità di componenti di tipo completamente diverso, detti FPGA (Field Programmable Gate Array [6]) nei quali viene messa a disposizione una matrice di *celle* ognuna delle quali può implementare o una semplice rete combinatoria (di pochi ingressi da 1bit, normalmente qualche unità e che produce un'uscita di un singolo bit) o un piccolo registro capace di memorizzare un singolo bit. La funzionalità delle singole celle, così come i collegamenti fra le diverse celle nella matrice, possono essere completamente riconfigurati "scrivendo" valori opportuni nel controller della FPGA. Oltre a contenere decine o centinaia di migliaia di celle, le FPGA più moderne contengono anche un certo numero di piccoli blocchi di memoria e di DSP (Digital Signal Processors, dispositivi in grado di compiere semplici operazioni aritmetiche in virgola mobile) sparsi fra le celle e configurabili per interagire con le celle stesse. I blocchi di memoria possono essere utilizzati per mantenere risultati intermedi delle

computazioni senza doverli necessariamente trasferire su moduli di memoria esterni. I DSP permettono di svolgere velocemente operazioni in virgola mobile senza dover impegnare grosse quantità di celle riconfigurabili. Utilizzando FPGA si possono “programmare” componenti hardware che, pur a velocità nettamente inferiori a quelle che si possono ottenere con un processo di produzione di chip classico, calcolano molto velocemente funzioni particolari. Per esempio, le FPGA vengono utilizzate per l’accelerazione dell’esecuzione di algoritmi crittografici particolarmente pesanti se eseguiti sulla CPU. Il vantaggio dell’uso di questi dispositivi è che se in momenti diversi dobbiamo eseguire calcoli diversi, l’FPGA può essere riprogrammata *on-the-fly* in tempi che sono dell’ordine dei micro o millisecondi.

Entrambi questi tipi di dispositivi sono stati inizialmente messi a disposizione come componenti che si affacciano sul bus del processore, e dunque interagiscono con la CPU con i meccanismi tipici delle unità di ingresso/uscita. La loro programmazione richiede ancora una volta l’orchestrazione esplicita di un insieme di flussi di controllo diversi come nel caso dei multi-core. Ma mentre i multi-core mettono a disposizione del programmatore un certo numero di risorse di calcolo completamente omogenee fra di loro e quindi in grado di eseguire tutte lo stesso insieme di compiti appartenenti ad un certo programma, FPGA e GP-GPU mettono a disposizione del programmatore modalità molto efficienti per eseguire tipi ben particolari di computazioni. In aggiunta a questo, mentre per programmare i core di un’architettura multi-core si possono utilizzare linguaggi e strumenti di programmazione classici, per la programmazione di GP-GPU ed FPGA è necessario adottare linguaggi e strumenti di programmazione *ad hoc*, spesso di non facile utilizzo per i programmatori che si sono formati utilizzando ambienti di programmazione classici.

Inoltre, la tendenza più recente è quella di integrare direttamente GP-GPU e FPGA sullo stesso chip del processore multi-core mettendo a disposizione i cosiddetti *system on chip* (SOC). Le GP-GPU sono già parte integrante dei processori dei maggiori produttori mondiali, sia di quelli dedicati al mercato consumer che di quelli dedicato al mercato high end e HPC (High Performance Computing). Diversi produttori stanno già proponendo modelli con FPGA integrate sul processore, da utilizzare per l’accelerazione di quei compiti per i quali non esiste un’implementazione efficiente o abbastanza veloce su CPU o GP-GPU. Un esempio di SOC che integra multi-core e FPGA è il sistema ZYNQ della Xilinx®, che integra sullo stesso chip due core ARM®, una FPGA che può contenere da 28K a 444K celle riconfigurabili e tutta la logica di interconnessioni e di supporto necessaria.

## 2. Lo scenario software

Le CPU multi-core rendono dunque lo scenario che si presenta al programmatore di applicazioni uno scenario decisamente parallelo. Applicazioni che sfruttino efficientemente un’architettura multi core devono necessariamente mettere a disposizione più attività parallele (flussi di controllo) per permettere di mantenere occupati i diversi core disponibili. Le GP-GPU rendono questo

0

1

0

1

0

scenario ulteriormente parallelo, visto il numero di core presenti sulle GP-GPU, ma anziché richiedere generiche attività parallele, richiedono attività che calcolano, in parallelo, lo stesso tipo di funzione su tutti i componenti di un certo dato. GP-GPU e FPGA richiedendo entrambe tecniche piuttosto particolari di programmazione il che rende lo scenario decisamente eterogeneo. Ma quali sono gli strumenti a disposizione del programmatore per realizzare applicazioni in grado di sfruttare al meglio le caratteristiche di questi nuovi dispositivi di calcolo?

Per realizzare applicazioni su queste architetture sono disponibili svariati tipi di modelli e strumenti di programmazione, ed in particolare ogni tipo di architettura ha i propri strumenti.

Su CPU multi-core gli standard di fatto sono OpenMP [7,8], per i sistemi multi core a memoria condivisa e MPI [9], per le reti/cluster di calcolatori. OpenMP è un modello di programmazione *ad ambiente globale*, ovvero nel quale si assume che tutte le attività che avvengono in parallelo nel corso dell'esecuzione di un programma abbiano accesso ad un livello di memoria condivisa. MPI invece è un modello di programmazione a *scambio di messaggi* (o *ad ambiente locale*), ovvero nel quale si assume che tutte le attività parallele all'interno del programma abbiano accesso solo ad una propria memoria locale ed eventuali dati da condividere debbano essere esplicitamente spediti alle attività che li condividono. Da un altro punto di vista, OpenMP richiede la sola introduzione nel codice sequenziale di *annotazioni* che creano e orchestrano le attività parallele. La compilazione tradizionale del codice OpenMP ignora le annotazioni ed esegue dunque il codice sequenzialmente, senza che il programmatore debba modificare alcunché nel sorgente. Per contro, MPI richiede l'introduzione di un discreta quantità di codice vero e proprio nell'applicazione, secondo un modello SPMD (Single Program Multiple Data [10]). Sostanzialmente il programmatore deve fornire esplicitamente il codice per tutte le attività che verranno eseguite in parallelo, comprensivo delle istruzioni per la comunicazione e sincronizzazione delle informazioni da condividere fra le varie attività parallele.

Su GP-GPU abbiamo OpenCL [11] e CUDA [12] (quest'ultimo disponibile solo sulle GP-GPU nVidia) anche se recentemente stanno prendendo piede anche standard leggermente di più alto livello come AAC. Entrambi i modelli espongono al programmatore un livello di astrazione molto basso. Concetti come la gestione dell'allocazione dei dati in memoria, che sono completamente nascosti al programmatore di applicazioni sequenziali da decenni, qui tornano ad essere sotto la responsabilità del programmatore delle applicazioni e determinano in modo significativo le performance che si riescono ad ottenere dalla GPU. Lo standard AAC [13] cerca di ridurre lo sforzo di programmazione richiesto, utilizzando annotazioni nello stile utilizzato da OpenMP per le CPU, ma ancora al prezzo di prestazioni sensibilmente inferiori rispetto a quelle ottenibili dall'uso diretto di OpenCL o CUDA.

Infine, sulle FPGA lo standard di fatto è rappresentato dall'utilizzo di linguaggi come VHDL [14] e Verilog [15], che sono veri e propri *linguaggi per la descrizione dell'hardware* (HDL) e che richiedono una profonda conoscenza

delle tecnologie di realizzazione delle FPGA per poter essere utilizzati efficientemente. In Verilog o VHDL si descrivono una serie di componenti hardware e collegamenti fra i componenti in termini delle funzionalità elementari messe a disposizione dalle celle delle FPGA e successivamente un compilatore provvede a trasformare queste descrizioni nei file di configurazione da caricare nelle memorie di controllo delle FPGA. Il procedimento di compilazione, fra l'altro, è estremamente complicato e richiede la soluzione di diversi problemi molto complessi e non è raro che richieda ore per il completamento. Recentemente, si stanno muovendo i primi passi verso l'adozione, anche sulle FPGA, di OpenCL (Altera, CodePlay) così come di modelli di programmazione più evoluti e ad alto livello come Chisel [16], che introduce la tecnologia Object Oriented per descrivere componenti hardware e successivamente compila il codice a oggetti in Verilog o VHDL.

Al di là degli standard *di fatto* che vengono utilizzati sui diversi tipi di componenti, vanno citati tutta una serie di ambienti di programmazione in grado di supportare lo sviluppo di applicazioni parallele, in grado di utilizzare al meglio almeno le architetture multi e many core di tipo general purpose e, in qualche caso, anche le architetture con GPU. Questi modelli di programmazione sono vari e offrono livelli di astrazione e meccanismi per le computazioni parallele diversi al programmatore di applicazioni. Per esempio:

- i modelli basati su *PGAS* (Partitioned Global Address Space [17]) espongono al programmatore uno spazio di memoria comune partizionato fra i diversi esecutori dei flussi di controllo. I linguaggi di programmazione Unified parallel C [44], Fortress [45], X10 [46] e Global Array [44] fanno tutti parte di questa categoria;
- i modelli basati su *fork/join* (operazione di creazione di un flusso di controllo indipendente e di attesa esplicita della sua terminazione) espongono al programmatore la possibilità di lanciare computazioni parallele arbitrarie ma indipendenti e di attenderne la terminazione quando i loro risultati risultino necessari per il prosieguo dell'applicazione. Cilk [18] e la sua variante Cilk++ [19] sono linguaggi che fanno parte di questa categoria, così come Erlang [20]. Il già citato OpenMP, Intel TBB [25] e la Task Parallel Library di Microsoft [26] sono ambienti di programmazione che di fatto adottano questo modello;
- i modelli basati su *mapreduce* (esecuzione di una funzione su tutti gli elementi di una grossa collezione di dati e calcolo di una seconda funzione, binaria, associativa e commutativa su tutti i risultati prodotti dalla prima funzione) danno al programmatore la possibilità di esprimere in forma molto sintetica una vasta classe di funzioni su moli di dati di grosse dimensioni, dal calcolo delle occorrenze di una stringa in un insieme di testi al ranking delle pagine utilizzato da Google nei suoi motori di ricerca. Il framework di ricerca e classificazione delle informazioni di Google [21] e il framework Hadoop [22], che ne rappresenta la versione open source, fanno parte di questa categoria;

- i modelli basati su *BSP* (Bulk synchronous parallel) permettono ai programmatori di programmare attività parallele organizzate in *superstep*, passi in cui per un certo tempo un insieme di flussi di controllo procedono in maniera completamente indipendente alla fine dei quali si ha una fase di scambio di informazioni ottimizzata e bloccante fra tutti i flussi di controllo. A questa categoria appartengono Pregel [23] e Python BSP [24].

Ognuno di questi modelli di programmazione (e ce ne sono molti altri, in realtà, a partire da quelli object oriented fino a quelli puramente funzionali) offrono modalità diverse per programmare applicazioni parallele, ma tutti però condividono una caratteristica: mettono a disposizione del programmatore o strumenti di basso livello che il programmatore deve utilizzare per implementare le astrazioni di più alto livello di cui ha bisogno (è il caso di TBB, per esempio), o meccanismi di più alto livello ma adatti a modellare solo certe forme di calcolo parallelo (è il caso del modello *mapreduce*, per esempio) e dunque difficili da utilizzare qualora si vogliano utilizzare forme di calcolo parallelo diverse.

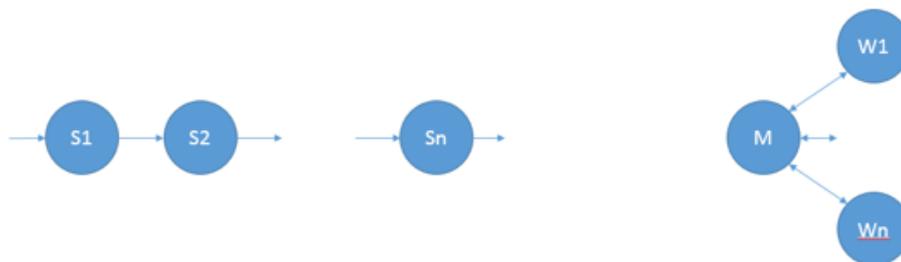
### 3. Pattern per la programmazione parallela

Fin dagli anni '90 ci si è resi conto che molte applicazioni parallele, programmate utilizzando framework di programmazione diversi, in realtà utilizzavano meccanismi di computazione parallela simili nel modo in cui le diverse attività parallele venivano create e fatte interagire fra di loro.

Ad esempio, in molti casi le computazioni parallele sono organizzate in *stadi* secondo il modello *pipeline* già menzionato a proposito dei modelli di processore nella sezione 1.1 [27]. Le applicazioni di video processing utilizzano spesso stadi in cascata per filtrare e correggere il rendering dei singoli frame di un filmato. Alcune applicazioni matematiche utilizzano lo stesso schema di calcolo con stadi che calcolano risultati parziali sempre più vicini al risultato finale dell'applicazione. In questo caso, il *pattern* di calcolo parallelo si può rappresentare come in Fig. 1 (parte sinistra) con una serie di attività concorrenti  $S_1, \dots, S_m$  dove l'attività  $S_i$  riceve i propri dati di input dall'attività  $S_{i-1}$  e invia i propri risultati all'attività  $S_{i+1}$ . L'attività  $S_1$  legge i dati in input dall'esterno e l'attività  $S_m$  spedisce all'esterno i propri risultati. Il pattern è utile solo nel caso in cui si debbano calcolare una serie di risultati corrispondenti a una serie di dati in input: serie di frame per il video processing, serie di dati strutturati (matrici o vettori) per le applicazioni matematiche. In questo caso, la computazione dei vari stadi  $S_i$  può avvenire in parallelo su dati relativi a ingressi diversi.

Un altro pattern tipico è quello detto *master-worker* ([28] detto anche *task farm* o semplicemente *farm*, vedi Fig. 1 (parte destra). In questo caso un'attività *master* deriva una serie di dati (detti *task*) sui quali va effettuata una certa computazione  $f$  o da una serie di dati in ingresso o da un unico dato complesso, e distribuisce l'esecuzione dei vari task a un insieme di  $m$  attività *worker* in grado di eseguire il calcolo di  $f$  sui task e restituire al master i risultati. Il calcolo della moltiplicazione di due matrici può essere organizzato secondo uno schema master-worker in cui il master manda ai worker coppie formate da righe della prima matrice e colonne della seconda e i worker calcolano il prodotto vettore-vettore restituendo un singolo elemento della matrice risultato al master.

In un ambito completamente diverso, un pattern master-worker può essere utilizzato per svolgere computazioni di reti neurali, con il master che assegna ai worker sotto partizioni della rete neurale per la computazione e riceve dai worker i valori calcolati da propagare verso altre partizioni o da utilizzare per iterazioni successive del calcolo della rete neurale.



**Figura 1**  
*Pattern pipeline (a sinistra) e master/worker (a destra).*  
*I cerchi indicano attività concorrenti e le frecce le comunicazioni fra attività.*

### 3.1 Algorithmic skeleton

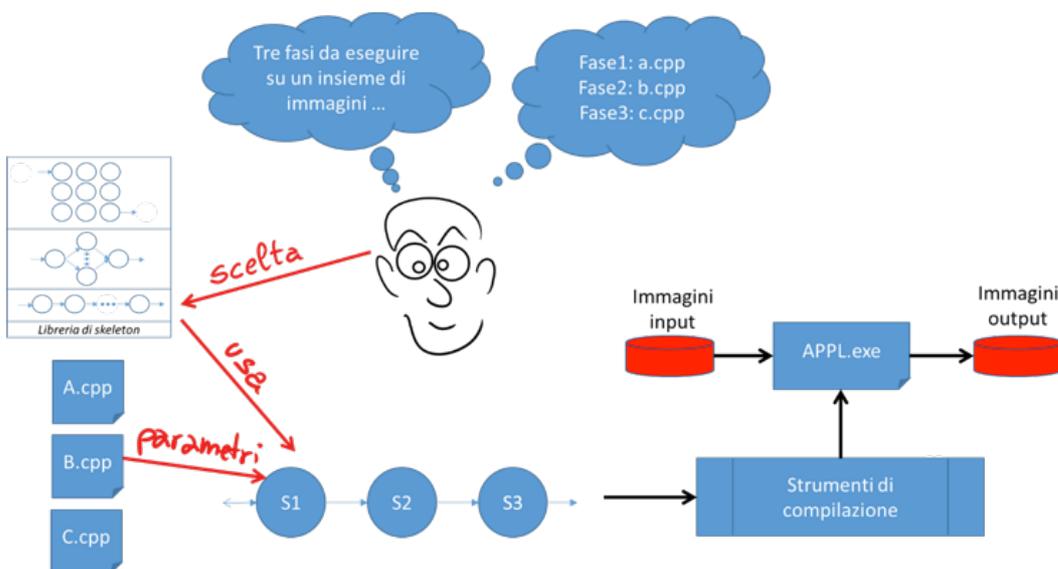
Riconosciuto questo fatto, si è cercato di implementare ambienti di programmazione che mettessero a disposizione astrazioni in grado di modellare completamente questi pattern:

- in *forma parametrica*, per poter specializzare quale computazione calcolare, per esempio specificando cosa calcolare negli stadi di un pipeline o nei worker del master-worker;
- realizzati in modo *efficiente*, sfruttando al meglio i meccanismi di supporto a basso livello;
- in modo che possano essere liberamente composti per ottenere forme di parallelismo più complesse (per esempio pipeline con stadi paralleli organizzati secondo pattern master worker)
- e che mettessero a disposizione del programmatore di applicazioni tutti i pattern più comunemente utilizzati per programmare applicazioni parallele.

Ambienti simili sono stati sviluppati fin dai primi anni '90 e sono noti come ambienti basati su *algorithmic skeleton* dal nome dato ai costrutti che implementano i pattern paralleli da Murray Cole nella sua tesi di dottorato [29] e nel successivo “manifesto” della programmazione parallela strutturata [30].

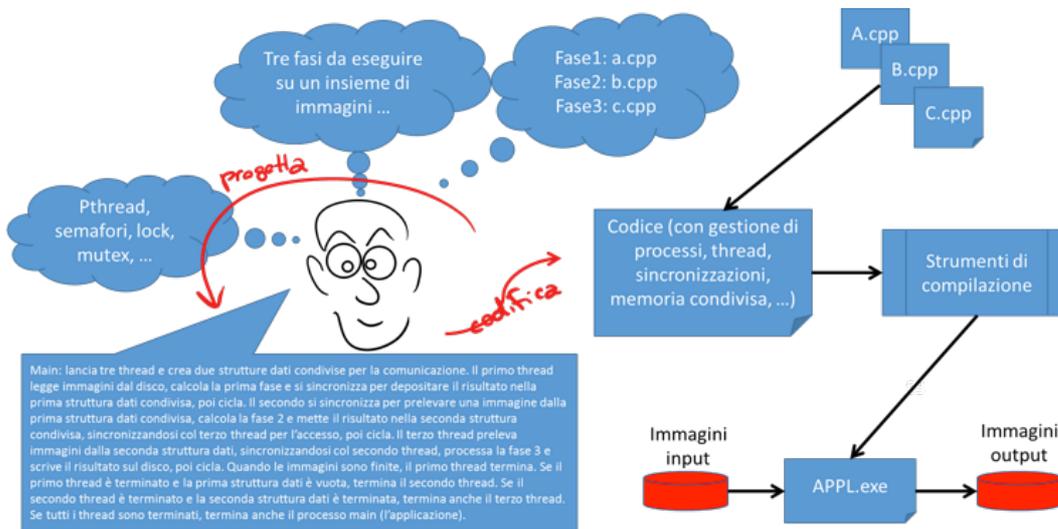
Sviluppare applicazioni parallele quando sia disponibile uno di questi ambienti vuol dire essenzialmente seguire un procedimento in fasi (vedi Fig. 2):

- capire quali pattern si vuole/devono utilizzare per modellare il parallelismo all'interno dell'applicazione
- istanziare gli algorithmic skeleton corrispondenti con il codice necessario ad implementarne le parti sequenziali (stadi di un pipeline o worker di un farm)
- verificare i risultati dell'esecuzione dell'applicazione ed eventualmente ripetere il processo utilizzando skeleton o composizioni di skeleton diverse fino a che non si raggiungono risultati ragionevoli in termini di prestazione (performance tuning).



**Figura 2**  
 Realizzazione di applicazioni parallele utilizzando un framework basato su algorithmic skeleton

Questo procedimento è molto diverso da quello utilizzato quando si programmano applicazioni parallele utilizzando strumenti di programmazione tradizionali e di basso livello (thread con primitive di sincronizzazione, processi con primitive di comunicazione e sincronizzazione, etc.). In questo caso, infatti, oltre ad identificare le forme di parallelismo necessarie per implementare la propria applicazione e a scrivere il codice per la *business logic* dell'applicazione, occorre anche implementare tutto ciò che serve per mettere a disposizione queste forme di parallelismo utilizzando i meccanismi di supporto a basso livello disponibili (vedi Fig.3).



**Figura 3**  
Realizzazione di un'applicazione parallela utilizzando framework non strutturati.

L'utilizzo di composizioni di skeleton per implementare applicazioni parallele e la conseguente conoscenza esplicita, a livello degli strumenti di compilazione e di run time, di ogni dettaglio relativo al parallelismo, ha diverse conseguenze:

- al programmatore dell'applicazione non è più richiesta alcuna conoscenza riguardo le caratteristiche dell'architettura su cui andrà ad eseguire l'applicazione né tantomeno dei meccanismi e delle tecniche necessarie per implementare efficientemente pattern di calcolo parallelo. Queste conoscenze saranno invece richieste a chi implementa gli skeleton. Si realizza dunque una completa e vantaggiosa *separazione delle responsabilità*: il programmatore "applicativo" è responsabile della correttezza del codice che calcola il risultato dell'applicazione. Il programmatore "di sistema" è responsabile della correttezza ed efficienza del codice che implementa i pattern paralleli al programmatore delle applicazioni mediante gli algorithmic skeleton. Entrambi sfruttano al meglio solo le loro specifiche capacità e conoscenze.
- per ognuno degli skeleton utilizzati dal programmatore applicativo si possono scegliere le implementazioni a disposizione per gli strumenti di supporto a basso livello e per l'architettura su cui si farà girare l'applicazione. Portare un'applicazione su un'altra architettura richiederà dunque una semplice ricompilazione del sorgente che utilizza gli skeleton.
- si possono adottare, a livello di compilatore e run time, tutta una serie di ottimizzazioni relative all'implementazione efficiente di forme di computazione parallele. Tali ottimizzazioni includono tecniche automatiche di trasformazione di composizioni di skeleton in altre composizioni funzionalmente equivalenti ma con diverse caratteristiche di performance ed efficienza nell'utilizzo delle risorse, che altrimenti sarebbero ancora a carico del programmatore applicativo.

- si possono infine utilizzare, in modo automatico, eventuali acceleratori disponibili (per esempio GPU) per realizzare in maniera più veloce o meno costosa in termini di potenza consumata tutti quegli skeleton per i quali esistono implementazioni sia per CPU che per gli acceleratori a disposizione [42, 43].

### 3.2 Parallel design pattern

L'idea di algorithmic skeleton ha portato allo sviluppo di un certo numero di ambienti di programmazione fra i quali ricordiamo gli ambienti FastFlow [32,33], Muesli [34], SKEPU [35], Sketo [36] e OSL [37]. Più o meno negli stessi anni però, in una comunità quasi completamente disgiunta rispetto alla comunità dell'high performance computing che ha dato vita agli algorithmic skeleton, ovvero nella comunità che si occupa di software engineering, è stato sviluppato un concetto molto simile a quello degli algorithmic skeleton: quello di *parallel design pattern*. Un *design pattern* descrive una metodologia di programmazione (per lo più object oriented) tipica ed utilizzata in molti contesti diversi, dandole un nome, discutendone le caratteristiche, gli usi tipici e le tipiche modalità di implementazione. Ad esempio, il design pattern chiamato *proxy* definisce la metodologia di programmazione in cui si usa un oggetto per filtrare le richieste dirette ad un altro oggetto. E' la metodologia utilizzata per esempio sui server web per far sì che le richieste dirette ad un server vengano invece processate da un altro server che le filtra e le inoltra ai server cui sono veramente dirette, tipicamente per implementare una qualche forma di controllo degli accessi. Un *parallel design pattern* definisce, analogamente, una metodologia di programmazione parallela, dandole un nome, discutendone le caratteristiche, etc. Tipici esempi di parallel design pattern (a diversi livelli di astrazione) sono il *dividi&conquista*, il *master/worker*, il *pipeline*.

A differenza degli algorithmic skeleton, i parallel design pattern *non mettono a disposizione del programmatore astrazioni tipiche del modello di programmazione* (chiamate di libreria, oggetti, funzioni di ordine superiore) che implementano direttamente il pattern parallelo. Piuttosto i parallel design pattern rappresentano delle vere e proprie *ricette* da utilizzare per implementare i pattern paralleli nelle applicazioni.

Ciò nonostante, i parallel design pattern hanno riscosso un successo anche maggiore di quanto non abbiano riscosso gli algorithmic skeleton. Gli esperti americani che hanno realizzato il celebre Berkeley report [38] hanno chiaramente identificato i parallel design pattern come il mezzo più adatto a formare una nuova classe di programmatori capaci di sviluppare rapidamente applicazioni parallele efficaci per le architetture parallele del nuovo millennio, auspicando adozione di approcci sintetizzabili come:

---

architecting parallel software with design patterns, not just parallel programming languages.

---

Inoltre, alcune grandi industrie hanno abbracciato il modello dei parallel design pattern, mettendo a disposizione librerie di componenti parallele utilizzabili per implementare parallel design pattern e dunque, in ultima istanza, di realizzare rapidamente efficienti applicazioni parallele. E' il caso dei Parallel design pattern di Intel® costruiti sopra alla loro libreria TBB o della Task Parallel Library di Microsoft, che lavora su C#.

### 3.3 Un buon risultato è sempre frutto di un buon compromesso

Parallel design pattern e algorithmic skeleton (di cui vediamo le principali caratteristiche nel riquadro 1) hanno entrambi vantaggi e svantaggi:

- i parallel design pattern aiutano il programmatore a capire esattamente cosa succede quando si implementa una certa forma di parallelismo, ma non mettono a disposizione un modo diretto per sperimentare pattern diversi in una certa applicazione, visto che di fatto non mettono a disposizione delle vere e proprie implementazioni;
- gli algorithmic skeleton, per contro, permettono di realizzare rapidamente un'applicazione quando esistano uno o più skeleton (o composizioni di skeleton) che implementano il pattern parallelo che il programmatore vuole implementare.

E' evidente che un ambiente di programmazione che possa presentare almeno in parte i vantaggi di entrambi gli approcci senza pagarne tutti gli svantaggi costituirebbe un grosso risultato in termini di modelli di programmazione per architetture parallele. In particolare, sarebbe molto produttivo avere un modello in cui il programmatore possa ragionare in termini di parallel design patterns in fase di progetto della propria applicazione per poi utilizzare algorithmic skeleton, anziché librerie/linguaggi a più basso livello di astrazione, per la realizzazione del pattern (o della composizione di pattern) scelta.

Recentemente, sono stati finanziati diversi progetti nei quali si è cercato di adottare questo approccio, fra i quali ricordiamo alcuni recenti progetti europei *ParaPhrase* (2012-2105 [40]) e *REPARA* (2013-2016 [41]), entrambi finanziati nel programma quadro FP7, e *RePhrase* (2015-2018), finanziato nel programma H2020. Tutti e tre questi progetti adottano in qualche modo un approccio combinato *parallel design patterns + algorithmic skeletons*, con diverse sfumature e obiettivi.

*ParaPhrase* ha avuto come obiettivo lo sviluppo di una metodologia di programmazione e dei relativi strumenti di programmazione nella quale si utilizzano i parallel pattern per modellare l'esplicitazione del parallelismo e si utilizzano tecniche semiautomatiche per introdurre e modificare (composizioni di) skeleton in codice sequenziale C++ o Erlang al fine di parallelizzare l'applicazione o di ottimizzarne l'esecuzione parallela. *REPARA* si propone di mettere a disposizione strumenti automatici per introdurre parallelismo (sotto forma di istanze opportune di algorithmic skeleton) in codice C++ sequenziale arricchito con annotazioni che identificano le possibilità di trasformazione del codice secondo parallel design pattern in modo da migliorarne sia i tempi di esecuzione che i consumi energetici (l'obiettivo è raddoppiare la velocità di esecuzione dimezzando il consumo energetico). In entrambi i progetti le

architetture target prese in considerazione includono CPU multi-core con acceleratori GPU e many core. REPARA considera anche la possibilità di utilizzare schede con FPGA. Sia per la parte C++ di ParaPhrase che per tutto il progetto REPARA, l'ambiente a skeleton utilizzato come run time è FastFlow [32] (vedi riquadro 2). RePhrase, infine, intende utilizzare parallel design pattern per l'orchestrazione delle attività parallele dell'applicazione e strumenti di programmazione di tipo classico (OpenMP, MPI, TBB) o basati su algorithmic skeleton (FastFlow) per l'implementazione dei pattern individuati nell'applicazione. All'interno del progetto verranno utilizzate massicciamente tecniche tipiche del software engineering sia per l'ottimizzazione dell'utilizzo dei design pattern che della loro implementazione in termini dei componenti messi a disposizione dagli ambienti a basso livello o di tipo algorithmic skeleton.

Alcuni risultati già disponibili dal progetto *ParaPhrase* hanno dimostrato che l'uso combinato dei pattern (per ragionare ad alto livello sulle forme di parallelismo da utilizzare nelle applicazioni) e degli algorithmic skeleton (per fornire al programmatore di applicazioni strumenti adatti ad implementare i parallel design pattern individuati) permettono la realizzazione di applicazioni parallele di efficienza comparabile o leggermente superiore rispetto ad analoghe applicazioni programmate con sistemi tradizionali ma con una drastica riduzione del tempo necessario a sviluppare e mettere a punto l'applicazione (ore invece che settimane per la parallelizzazione di tipiche applicazioni numeriche o di grafica computazionale) [40].

Va notato infine che esistono già diversi ambienti "di produzione" che utilizzando pesantemente la programmazione parallela strutturata, anche se in forme decisamente più limitate di quanto non venga fatto in questi progetti. Google ha fatto la sua fortuna su un framework che permette di realizzare velocemente efficaci applicazioni parallele che processano grandi quantità di dati secondo il pattern parallelo *MapReduce* (prima trasforma ogni dato in una coppia <chiave, risultato> poi "somma" tutti i risultati con la stessa chiave). Microsoft ha rilasciato la Task Parallel Library che permette di scrivere applicazioni parallele in cui il parallelismo è di fatto implementato opportune funzioni di libreria simili agli algorithmic skeletons. OpenCL, ambiente di programmazione utilizzato prevalentemente per le GPU, mette a disposizione del programmatore applicativo diversi pattern data parallel implementati come primitive del linguaggio (o algorithmic skeleton, se vogliamo).

#### 4. Conclusioni

I recenti ed importanti miglioramenti nelle tecnologie di produzione dei componenti di calcolo hanno prodotto macchine parallele ed eterogenee potenzialmente in grado di eseguire applicazioni con prestazioni molto elevate. Sfortunatamente, la programmazione di queste nuove macchine utilizzando gli strumenti di sviluppo classici richiede uno sforzo consistente ai programmatori applicativi, che si trovano a dover scrivere e gestire sia il codice necessario a implementare la *business logic* dell'applicazione che quello necessario a implementare e gestire la struttura parallela dell'applicazione stessa.

Per ovviare a questo problema, comunità di ricercatori appartenenti ad ambienti di ricerca diversi hanno prodotto nuovi strumenti di sviluppo basati su concetti di programmazione parallela strutturata. Questi ambienti di sviluppo permettono di semplificare la vita ai programmatori applicativi ma, al contempo, supportano lo sviluppo di applicazioni molto efficienti sulle nuove architetture disponibili.

In questo lavoro abbiamo introdotto brevemente le caratteristiche principali degli ambienti di programmazione basati su *algorithmic skeleton* e su *parallel design pattern*. Questi ambienti sono stati sviluppati principalmente dalle comunità di ricercatori che si interessano di *High Performance Computing* e *Software Engineering* ed hanno, al momento, una diffusione limitata ai gruppi di ricerca di università e grossi centri di ricerca. Diversi progetti, alcuni dei quali ancora in corso di svolgimento, hanno contribuito e contribuiscono al loro sviluppo e li stanno portando a livelli adatti alla loro adozione nel mondo dell'industria software. Dal canto loro, alcuni grossi nomi dell'industria informatica mondiale (fra cui Intel e Microsoft) hanno recepito parte dei concetti relativi alla programmazione parallela strutturata e li stanno adottando in alcuni dei loro prodotti.

A breve, ci dovremmo aspettare che queste tecnologie prendano il sopravvento, come accadde negli anni '80 nel mondo della programmazione sequenziale quando i linguaggi strutturati e object oriented presero il sopravvento sui linguaggi non strutturati usati fino a quell'epoca. Questo porterà ad uno sviluppo sempre più consistente degli strumenti di supporto alla programmazione parallela (compilatori, interpreti, librerie di run time) e ad un corrispondente sempre maggior livello di astrazione messo a disposizione ai programmatori applicativi. Questo permetterà di ridurre il *time-to-market* delle nuove applicazioni e lo sviluppo di applicativi sempre più complessi con un minore sforzo, in proporzione, da parte dei programmatori.

#### Riquadro 1: Algorithmic skeleton e parallel design pattern

	Algorithmic skeleton	Parallel design pattern
<b>Cosa sono</b>	Costrutti predefiniti, parametrici, che modellano forme di parallelismo note.	Descrizioni di forme di parallelismo (nome, uso tipico, problemi di implementazione, esempi di codice)
<b>Chi li promuove</b>	Comunità HPC	Comunità Sw Engineering
<b>Quando sono stati introdotti</b>	Inizio anni '90	Primi anni 2000
<b>Pubblicazione chiave</b>	M. Cole, "Bringing skeletons out of the closet: a pragmatic manifesto for skeletal parallel programming", Journal Parallel Computing Volume 30 Issue 3, March 2004 Pages 389 - 406	T. G. Mattson, B. A. Sanders, B. L. Massingill, Patterns for parallel programming, Addison Wesley, 2013, ISBN-13: 978-0321940780
<b>Framework che li adottano</b>	FastFlow, Muesli, Sketo, SKEPU, OSL	INTEL TBB Building Block Design Patterns White paper
<b>Come sono messi a disposizione</b>	Costrutti del linguaggio sequenziale ospite (chiamate di libreria, oggetti, funzioni higher order)	Descrizione (solo testo)

**Riquadro 2: FastFlow**

Cos'è:	Framework di programmazione parallela C++ che mette a disposizione come oggetti primitivi astrazioni che modellano i più comuni pattern paralleli (algorithmic skeletons)
Com'è fornita:	Libreria HPP-only: header files da includere e compilare insieme al codice applicativo
Per quali architetture:	CPU multi-core e many core (Intel, AMD, ARM, Tiler), GPU (nVidia CUDA, AMD OpenCL), reti di multi/many core con GPU (Ethernet, Infiniband)
Requisiti:	Compilatore C++11, libreria pthread
Quali pattern:	stream parallel (pipeline, task farm), data parallel (parallel for, map, reduce, stencil), high level (divide&conquer, pool evolution, macro data flow)
Scenari applicativi:	parallelismo a grana fine su architetture shared memory e cache coherent, applicazioni streaming, uso congiunto di multi-core e acceleratori
Use cases portfolio:	Bowtie2 porting, two phase video image restoring (CPU+GPU), Yadt C4.5 classifier, pbzip2 porting, block-based Cholesky & LU decomposition, ...
Licenza:	Open Source LGPL
Download:	svn checkout svn://svn.code.sf.net/p/mc-fastflow/code/ mc-fastflow-code
Sito web:	<a href="http://calvados.di.unipi.it/fastflow">http://calvados.di.unipi.it/fastflow</a>
Sample code:	Codice necessario per l'implementazione di un master/worker con NW workers:
	<pre> using namespace ff;  //calcolo di un task struct Worker: ff_node&lt;float&gt; {     float *svc(float *task) { .... }     ... op }; ... int main() {     ...     std::vector&lt;std::unique_ptr&lt;ff_node&gt;&gt; W;     for(size_t i=0;i&lt;NW;++i)         W.push_back(make_unique&lt;Worker&gt;());     ff_Farm&lt;&gt; unFarm(W);     if (unFarm.run_and_wait_end(&lt;0)         error("running unFarm");     ... }                 </pre>

### Riquadro 3: Sigle utilizzate nel testo

SIGLA	TP
BSP	Bulk Synchronous Computing
CPU	Central Processing Unit
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
GPU	Graphics Processing Unit
GP-GPU	General Purpose Graphics Processing Unit
HPC	High Performance Computing
MPI	Message Passing Interface
PGAS	Partitioned Global Address Space
SOC	System On Chip
SPMD	Single Program Multiple Data
TBB	Thread Building Blocks
TPL	Task Parallel Library

### Bibliografia

- [1] R. R. Schaller, "Moore's law: past, present, and future", *IEEE Spectrum*, Volume 34 Issue 6, June 1997, Page 52-59, IEEE Press
- [2] C. A. Mack, "Fifty Years of Moore's Law", *Semiconductor Manufacturing, IEEE Transactions on* Vol. 24 , Issue 2, pp 202-207, 2011
- [3] D. Ivosevic, N. Frid, "Performance-Occupation trade-off examination in custom processor design", *Proceedings of MIPRO 2014*, pp. 1024-1029, DOI: <http://dx.doi.org/10.1109/MIPRO.2014.6859719>, 2014
- [4] J. Jeffers, J. Reinders, Intel Xeon PHI Coprocessor High Performance Computing, Morgan Kaufmann, 2013, ISBN: 9780124104945
- [5] K. Fatahalian, M. Houston, "A Closer Look at GPUs", *Communications of the ACM*, Vol. 51 No. 10, Pages 50-57, 2008, DOI: 10.1145/1400181.1400197
- [6] I. Kuon, R. Tessier, J. Rose, "FPGA architecture: survey and challenges", *Foundations and Trends in Electronic Design Automation*, Vol 2, Issue 2, Feb 2008, pp. 135-253, Now Publishers

- [7] L. Dagum, R. Menon, "OpenMP: An Industry-Standard API for Shared-Memory Programming", *IEEE Computational Science & Engineering*, Vol. 5 Issue 1, Jan 1998, pp 46 – 55, IEEE Computer Society Press, DOI: 10.1109/99.660313
- [8] The OpenMP API specification for parallel programming, <http://openmp.org/wp/>, ultimo accesso febbraio 2015
- [9] W. Gropp, E. Lusk, A. Skjellum, Using MPI: Portable Parallel Programming with the Message-Passing Interface, 3<sup>rd</sup> edition, MIT Press, 2014.
- [10] F. Darema, "The SPMD Model: Past, Present and Future", Proceedings of the 8th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, Springer-Verlag London, 2001
- [11] B. Gaster, L. Howes, D. R. Kaeli, P. Mistry, D. Schaa, *Heterogeneous Computing with OpenCL*, 2011, Morgan Kaufmann
- [12] J. Nickolls, I. Buck, M. Garland, M. Skadron, "Scalable Parallel Programming with CUDA", *Queue*, Vol. 6 Issue 2, March/April 2008, pp. 40-53, ACM, DOI: 10.1145/1365490.1365500
- [13] OpenACC Directives for Accelerators, <http://www.openacc-standard.org/>, ultimo accesso, 2015
- [14] S. Mazor, P. Langstraat, *A guide to VHDL*, Springer Verlag, 2013, ISBN: 1475721161
- [15] S. Lee, *Advanced Digital Logic Design Using Verilog, State Machines, and Synthesis for FPGA's*, Thomson-Engineering, 2005, ISBN:0534551610
- [16] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avizienis, J. Wawrzyniek, K. Asanovic, "Chisel: constructing hardware in Scala embedded language", DAC '12 Proceedings of the 49th Annual Design Automation Conference, pp. 1216-1225, ACM, 2012, ISBN: 978-1-4503-1199-1
- [17] PGAS Partitioned Global Address Space, <http://www.pgas.org/>, ultimo accesso 2014
- [18] R. D. Blumofe, C. F. Joerg, B. C. Kuszmaul, C. E. Leiserson, K. H. Randall, Y. Zhou, "Cilk: an efficient multithreaded runtime system", PPOPP '95 Proceedings of the fifth ACM SIGPLAN symposium on Principles and practice of parallel programming, pp. 207-216, ACM, 1995
- [19] Intel, "Intel Cilk++ SDK Programmer's Guide", [https://www.clear.rice.edu/com422/resources/Intel\\_Cilk++\\_Programmers\\_Guide.pdf](https://www.clear.rice.edu/com422/resources/Intel_Cilk++_Programmers_Guide.pdf), 2009
- [20] F. Cesarini, S. Thompson, Erlang programming: a concurrent approach to software development, O'Reilly, 2009
- [21] J. Dean, S. Ghemawat, "MapReduce: simplified data processing on large clusters", CACM, Vol. 51, Issue 1, pp 107-113, ACM, 2008
- [22] T. White, Hadoop, the definitive guide, O'Reilly, 2009

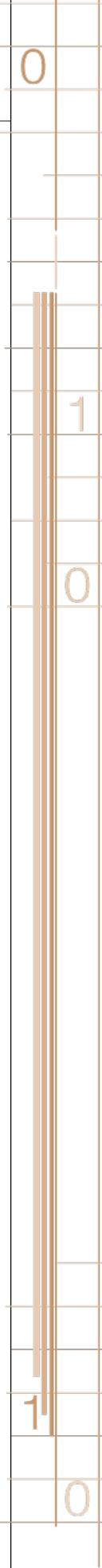
- [23] G. Malewicz, M. H. Austern, A. J.C. Bik, J. C. Dehnert, I. Horn, N. Leiser, G. Czajkowsky, "Pregel: a system for large scale graph processing", SIGMOD '10 Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pp. 135-146, ACM, 2010
- [24] K Hinsen, H P Langtangen, O Skavhaug, and A Odegard, "Using BSP and Python to simplify parallel programming", *Future Generation Computer Systems* 22(1-2), pp. 123-157, 2006
- [25] Intel, "Intel Thread Building Blocks", <https://www.threadingbuildingblocks.org/>, ultimo accesso, 2015
- [26] Microsoft, Task parallel library (TPL), <https://msdn.microsoft.com/it-it/library/dd460717%28v=vs.110%29.aspx>, ultimo accesso 2015
- [27] J. A. Pienaar, S. Chakradhar, A. Raghunathan, "Automatic generation of software pipelines for heterogeneous parallel systems", Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, IEEE Computer Society Press, 2012
- [28] J. Berthold, M. Dieterle, R. Logen, S. Priebe, "Hierarchical Master-Worker Skeletons", in *Practical Aspects of Declarative Languages*, LNCS, Vol. 4902, pp. 248-264, Springer Verlag, 2008
- [29] Murray I. Cole. Algorithmic Skeletons: Structured Management of Parallel Computation, MIT Press, 1989.
- [30] M. Cole, "Bringing skeletons out of the closet: a pragmatic manifesto for skeletal parallel programming", *Parallel Computing*, Vol. 30 Issue 3, pp. 389 - 406, 2004, Elsevier Science Publishers
- [31] K. Matsuzaki, K. Emoto, "Implementing fusion-equipped parallel skeletons by expression templates", Proceeding IFL'09 Proceedings of the 21st international conference on Implementation and application of functional languages, pp. 72-89, Springer Verlag, 2010
- [32] Fastflow home page, <http://calvados.di.unipi.it/fastflow>, ultimo accesso, marzo 2015
- [33] M. Aldinucci, M. Danelutto, P. Kilpatrick, M. Meneghin, M. Torquati, "Accelerating Code on Multi-cores with FastFlow", Euro-Par 2011 Parallel Processing, LNCS, Vol. 6853, pp. 170-178, Springer Verlag, 2011
- [34] S. Ernsting, H. Kuchen, "Algorithmic skeletons for multi-core, multi-GPU systems and clusters", *International Journal of High Performance Computing and Networking*, vol. 7 Issue 2, April 2012, pp. 129-138, Inderscience Publishers
- [35] J. Enmyren, C. W. Kessler, SkePU: a multi-backend skeleton programming library for multi-GPU systems, Proceedings of HLPP '10 Proceedings of the fourth international workshop on High-level parallel programming and applications, pp. 5-14, ACM, 2010
- [36] SkeTo project, <http://sketo.ipl-lab.org/>, ultimo accesso 2015

- [37] N. Javed, F. Loulergue, "OSL: Optimized Bulk Synchronous Parallel Skeletons on Distributed Arrays"; *Advanced Parallel Processing Technologies*, LNCS, Vol. 5737, pp. 436-451, Springer Verlag, 2009
- [38] K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiawicz, N. Morgan, D. Patterson, K. Sen, J. Wawrzynek, D. Wessel, K. Yelick, "A View of the Parallel Computing Landscape", *Communications of the ACM*, Vol. 52 No. 10, Pages 56-67
- [40] ParaPhrase home page, <http://www.paraphrase-ict.eu/>, ultimo accesso 2015
- [41] REPARA home page, <http://parrot.arcos.inf.uc3m.es/wordpress/>, ultimo accesso 2015
- [42] T. Serban, M. Danelutto, P. Kilpatrick, "Autonomic scheduling of tasks from data parallel patterns to CPU/GPU core mixes", *High Performance Computing and Simulation (HPCS)*, pp. 72-79, ISBN 978-1-4799-0836-3, IEEE Press, 2013
- [43] M. Goli, H. González-Vélez, "Heterogeneous Algorithmic Skeletons for Fast Flow with Seamless Coordination over Hybrid Architectures", *Proceedings of PDP 2013 Euromicro International Conference on Parallel, Distributed and Network based processing*, pp. 148-156, IEEE Computer Society Press, 2013
- [44] C. Coarfa, Y. Dotsenko, J. Mellor-Crummey, F. Cantonnet, T. El-Ghazawi, A. Mohanti, Y. Yao, and D. Chavarra-Miranda. "An evaluation of global address space languages: co-array fortran and unified parallel C", In *Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming (PPoPP '05)*, ACM, 2005
- [45] G. L. Steele, Jr.. 2006. "Parallel programming and code selection in fortress", in *Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel programming (PPoPP '06)*, ACM, 2006
- [46] P. Charles, C. Grothoff, V. Saraswat, C. Donawa, A. Kielstra, K. Ebcioglu, C. von Praun, and V. Sarkar, "X10: an object-oriented approach to non-uniform cluster computing", *SIGPLAN*, No. 40, 10 (October 2005)

## Biografia

**Marco Danelutto** ha ottenuto il Dottorato di Ricerca in Informatica nel 1990, è attualmente professore associato presso il Dipartimento di Informatica dell'Università di Pisa ed ha recentemente ottenuto l'abilitazione a professore di prima fascia. I suoi interessi di ricerca sono concentrati su modelli e strumenti per la programmazione parallela strutturata. E' autore di oltre 140 pubblicazioni internazionali su riviste e conferenze con peer review. Attualmente è presidente del Corso di Laurea Magistrale congiunto (Università di Pisa e Scuola Superiore Sant'Anna) in Informatica e Networking.

email: [marco.danelutto@unipi.it](mailto:marco.danelutto@unipi.it)



# Le radici concettuali del computer

Luigi Borzacchini

## Sommario

*La computer science appare quasi senza storia, anche per le persone di cultura solo un caleidoscopio di magnifiche novità. Essa in realtà ha radici antiche ed emerge nell'Ottocento in primo luogo dalla tradizione matematica inglese, in secondo luogo da quella tedesca, che ne forniscono due diverse impostazioni, ancora oggi riconoscibili.*

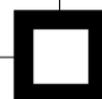
## Abstract

*Computer science seems today almost historyless, even for the intellectuals just a kaleidoscope of wonderful novelties. Actually it has deep historical roots and emerges in the 19th century first and foremost from the English mathematics tradition, and second from the German one, that gave it two different approaches, even today recognizable.*

**Keywords:** History of computer science, 19th century English Mathematics, 19th century German Mathematics, Boole and the Algebraic Logic, Babbage, Lady Lovelace, Hilbert, Turing.

## 1. Introduzione

Qualche anno fa decisi di tenere, in via sperimentale, agli studenti di informatica un corso specialistico a carattere storico, una "storia del pensiero logico ed algoritmico", dai Babilonesi a Steve Jobs. L'accoglienza da parte degli studenti fu abbastanza positiva ma le restrizioni sul numero dei corsi portarono dopo qualche anno alla sua chiusura. All'inizio del corso chiedevo agli studenti: <ma siete sicuri di essere solo figli di un microchip e di una stampante?>, perché questa era l'impressione che mi dava la loro reazione iniziale alla idea stessa che la computer science potesse avere una storia che andasse al di là di una mezza paginetta su Pascal e Leibniz. Sembravano accolti di una scienza esile e di una tecnologia imperiale, entrambe senza storia.



I computer non crescono sugli alberi e sono apparsi alla fine di una storia lunga e complessa, e la computer science è caratterizzata oggi (ma anche già in Turing) come qualcosa di più di un gruppo di tecniche e di una tecnologia per fare i calcoli meccanicamente: le abilità del professionista o dello scienziato del computer sono varie, ma ruotano sempre intorno al problema della traduzione in codice, della riduzione sintattica dei problemi reali concernenti il mondo che ci circonda o lo stesso computer, problemi di rappresentazione insomma [vedasi riquadro 1]. La computer science nasce quindi su quel versante della umana intelligenza che chiamiamo talora 'il pensiero formale', il pensare manipolando segni, il 'ragionare senza comprendere' - la lingua simbolica che, come diceva Frege, può "pensare per noi"- erede del fatto che l'uomo, a differenza di Dio che conosce direttamente tutto perché l'ha creato, conosce solo attraverso i segni.

Un esempio può chiarire il concetto. Se io dico <Giovanni è romano> e <Tutti i romani sono italiani>, ognuno si sente di concludere <Giovanni è italiano>, e magari lo conosce anche un tale Giovanni. Ma se io dico <Giovanni è un sarchiapone> e <Tutti i sarchiaponi sono sesquipedali>, alla fine siamo anche disposti a concedere che <Giovanni è sesquipedale>, ma con qualche perplessità, perché non sappiamo di chi e cosa stiamo parlando, ci siamo solo affidati alla deduzione formale che da <a è B> e <Tutti i B sono C> deduce <a è C>, Bene questo è un ragionamento puramente formale, un "pensiero cieco", secondo Leibniz, che non 'comprende' e non 'vede' ma semplicemente manipola segni.

E' un'idea antica, affermata definitivamente soprattutto nel Seicento nell'opera di Leibniz. E la 'ragione formale' è forse l'unica capacità intellettuale esclusiva dell'uomo: tra gli animali c'è qualche abilità linguistica, anche logica, ci sono abilità di problem solving, ed anche qualche capacità matematica sui numeri piccoli, ma nessuno ha mai visto un animale risolvere un problema manipolando segni.

Questa forma di pensiero è sempre apparsa nelle riflessioni di filosofi e uomini di scienza sino dalla antichità, è una capacità che è al centro di discipline quali matematica, logica, filosofia, etc, delineando un *regno dei segni* in continua espansione, del quale la computer science è stata l'esito finale e il computer ne è stato l'epifania, in una storia in cui esigenze puramente pratiche sono state il terreno di coltura di una teoria della calcolabilità la cui profondità sia scientifica che filosofica non è ancora oggi pienamente compresa. Possiamo dire che *la logica e la filosofia della conoscenza sono per il computer quello che la termodinamica è per il motore a scoppio.*

Negli ultimi venti anni ho scritto molto per rintracciare la genesi e l'evoluzione del pensiero formale dall'antichità al XIX secolo, ed è quindi inevitabile tracciarne oggi il profilo genetico.

## 2. Nella matematica moderna

Le radici immediate del computer sono nella matematica dell'Ottocento, un secolo in cui esistevano diverse tradizioni scientifiche in Europa. E' infatti una caratteristica della matematica moderna europea quella di essere cresciuta in diverse tradizioni nazionali, nate nel Rinascimento, quando da un lato la matematica si diffondeva nella vita economica e sociale, e dall'altro il latino perdeva la sua centralità a vantaggio dei nuovi idiomi nazionali: Galileo scriverà in italiano e Descartes in francese. D'altro canto però queste tradizioni, pur conservandosi, riuscivano a riconoscersi in una unica 'matematica': ad esempio, per quanto diverse le ispirazioni del calcolo differenziale di Leibniz e Newton, all'inizio del Settecento quel calcolo aveva trovato una sua struttura standard.



**Figura 1**  
*Alan M. Turing*

All'inizio dell'Ottocento si comincia a distinguere tra sintassi e semantica: la prima riguarda una matematica sempre più autonoma e astratta, la seconda riguarda invece l'applicazione della matematica alla fisica. La tradizione matematica italiana era ormai abbastanza marginale, soprattutto le erano rimasti abbastanza estranei i temi algebrici e simbolici: Galileo ignorava l'algebra e i suoi allievi avevano qualificato l'algebra simbolica come un 'gioco infantile'. Forse la scuola matematica più importante all'inizio dell'Ottocento era quella francese e francese era stata l'origine dell'algebra simbolica, con Descartes e Fermat. Ma la matematica francese era rimasta legata ad una connessione strettissima tra fisica e matematica, incapace di vedere la crescente autonomia della sintassi e del calcolo simbolico, e così non ebbe quasi nessun ruolo nel formarsi della computer science: ancora oggi i Francesi sono quasi gli unici al mondo a non chiamare 'computer' il computer!

L'attenzione al rapporto tra sintassi e semantica sarà invece diffuso nella matematica tedesca, ed infatti la matematica e la logica tedesca avranno un grande ruolo nella nascita della cultura informatica, soprattutto nei suoi aspetti linguistici e fondazionali della conoscenza, e il formalismo di Hilbert avrà un ruolo importante nella formazione di Turing, il cui maestro a Cambridge fu un hilbertiano, Max Newman: il famoso suo saggio del '36-'37 concerneva l'*Entscheidungsproblem* [1], cruciale problema hilbertiano [vedasi riquadro 2]. E lo stesso John von Neumann, a cui usualmente si ascrive l'architettura di base del computer moderno, era un hilbertiano. E questa importante influenza formalista è la trama di fondo su cui ritorneremo nel seguito, ma l'ordito, il tronco principale della storia del computer e della computer science, da Babbage (1792-1871) a Turing (1912-1954), affonda le sue radici nella cultura logico-matematica inglese e poi nord-americana. E, cosa ancora più importante, ne porta ancora oggi tutti i segni.



**Figura 2**  
**G. W. Leibniz**

Tutto era cominciato due secoli prima, nel Seicento, con la Rivoluzione Scientifica. La nascita dell'algebra simbolica e l'esplosione della idea del 'regno dei segni' come dimora della scienza ne era stato forse il processo fondamentale. E sin dall'inizio era stato un tema su cui convergevano l'empirismo e pragmatismo inglese e la filosofia razionalistica tedesca. Da un lato Hobbes e Locke, dall'altro Leibniz concepirono l'idea che *il pensiero sia calcolo*. Nei "Nuovi Saggi sull'Intelletto Umano" (1705) Leibniz critica Locke, ma in fondo ne condivide l'agenda dei problemi.

Il regno dei segni per Leibniz (1646-1716) come per la successiva matematica tedesca è il linguaggio della 'teoria', mentre per la matematica inglese è il 'calcolo'. Nella tradizione culturale empirista inglese il pensare è in primo luogo associazione di 'idee', che conserveranno al lungo la loro originale natura di 'immagini mentali' costruite induttivamente dall'esperienza sulle cose. Bene, l'idea che emerge tra gli empiristi è che il calcolo, anche se le idee sono eterogenee rispetto ai segni, può riflettere la associazione e separazione tra le 'idee'. Ha un ruolo solo pratico, i 'segni' nel mondo sono degli intrusi, e lo stesso Newton, padre del calcolo differenziale, per la sua fisica userà la rappresentazione geometrica, adoperando le tecniche differenziali solo per fare qualche 'calcolo', inventando a tal fine anche le tecniche per il calcolo approssimato.

Per Leibniz invece l'*ars combinatoria*, la scienza della manipolazione dei segni, è omogenea alla aritmetica nel costituire il linguaggio di base della scienza, ed ha due aspetti che sono ancora oggi alla base della nostra idea di 'linguaggio di rappresentazione': è una *characteristica universalis*, cioè un lessico in grado di tradurre in sequenza di segni qualsiasi fatto esprimibile, ed è un *calculus ratiocinator*, cioè una notazione simbolica in grado di ricavare tutte le inferenze e di descrivere tutti i calcoli. E nella computer science questi due aspetti del linguaggio di rappresentazione saranno ingredienti essenziali; infatti ad un qualunque linguaggio per data-base, o più in generale di rappresentazione della conoscenza, questo si richiede: da un lato che possa 'rappresentare' tutte le informazioni e nozioni disponibili, e dall'altro che lo si possa 'utilizzare' per ottenere algoritmicamente le informazioni e conseguenze utili.

Esempio classico della differenza fra le due tradizioni è il 'calcolo con le lettere' che studiamo nelle scuole medie: per gli inglesi è una 'aritmetica generale', la generalizzazione del calcolo con i numeri, per i tedeschi è una 'matematica universale' con da applicare al concetto generale di 'grandezza'.

Oppure il concetto di "numero reale", nato all'inizio del Seicento come numero che richiede infinite cifre decimali per essere scritto, come misura delle grandezze fisiche e come corrispettivo aritmetico del punto geometrico. In Inghilterra era apparso coi logaritmi, creati esplicitamente solo come tecnica per facilitare i calcoli, e inglese sarà il regolo calcolatore per fare i calcoli sui numeri

reali. In Germania invece verranno studiati nell'Ottocento il 'concetto' e la 'teoria' dei numeri reali.

In generale diverso è il 'mondo fisico' dai due punti di vista. Nel Continente appare retto da principi e leggi di conservazione o di massimo e minimo, creato da un Dio 'orologiaio' che dopo averlo creato lo ha lasciato muoversi e mutare secondo le sue proprie leggi. La meccanica continentale sarà fatta di 'principi' di conservazione (energia, quantità di moto, etc.) e di massimo (entropia, etc.), lasciando ai margini la concatenazione delle azioni causali. Quello newtoniano invece è un mondo di forze agenti e reagenti, opera di un Dio continuamente attivo, che di esso garantisce il divenire con catene causali governate dal soggetto, umano o divino. Anche nel nostro immaginario collettivo inglese è l'inventore, l'uomo pratico che ama il rischio e l'avventura, empirico e ingegnoso, l'imprenditore, mentre tedesco è l'uomo di scienza, teorico e metafisico, il professore severo e razionale che legge e scrive grossi libri. Mayr ha sottolineato come anche nella scienza politica il Continente restò legato alla metafora dello Stato come orologio, mentre l'Inghilterra preferì vederlo come una bilancia o un essere animato [2].

E sin dal Seicento, mentre in Leibniz la fisica-matematica assume una vocazione simbolica, la matematica inglese conserva l'opposizione tra un mondo reale da descrivere geometricamente ed un mondo di simboli utile solo come strumento per fare i calcoli. Anche nel Settecento il calcolo aritmetico nei testi inglesi avrà un carattere molto pratico e strumentale, ed il testo più teorico, l'"Aritmetica Universale" di Newton, vedrà il calcolo letterale come una 'astrazione' dell'usuale calcolo elementare. Nel frattempo i segni algebrici nell'analisi matematica del Settecento avevano progressivamente sul Continente completamente sostituito l'iniziale impianto geometrico newtoniano nella nuova fisica-matematica, adottando definitivamente il linguaggio simbolico che oggi troviamo nei nostri corsi di analisi matematica.

Così all'inizio dell'Ottocento tra i matematici inglesi cresce la consapevolezza che l'attaccamento al linguaggio geometrico era diventato una palla al piede, e nasce la *Cambridge Analytical Society* ad opera di alcuni matematici per promuovere il nuovo stile analitico. Così, mentre la matematica francese continua a coltivare il rapporto intimo tra matematica e fisica, e quella tedesca si tuffa nello studio di funzioni strane e paradossali per chiarire i concetti delle teorie matematiche, la matematica inglese sottolinea il suo carattere calcolistico e algebrico-simbolico [3].

Tra la matematica tedesca e quella inglese nell'Ottocento si delinea una sorta di antifona: gli assiomi sono per i primi la costruzione deduttiva della teoria, per i secondi sono le regole del calcolo, la semantica per i primi riguarda i modelli delle teorie, per i secondi l'interpretazione non aritmetica dei calcoli, e in tali 'interpretazioni' si intravede il soggetto empirico che vuole applicare il calcolo, mentre nei 'modelli' c'è l'universo dei mondi possibili della teoria, senza un soggetto.

Per i matematici inglesi il calcolo per molto tempo conserverà il cordone ombelicale con la sua originale natura aritmetica, che sarà la sua interpretazione standard e che però col tempo perderà la sua centralità, dando origine a nuovi

'calcoli'. E' l'approccio 'calcolistico' che permette di chiarire una differenza fondamentale nei linguaggi formali: la differenza tra operazioni e numeri si traduce in quella tra simboli 'primitivi', il cui significato è fisso e dato dalle regole del calcolo, e simboli 'interpretabili', la cui interpretazione rappresenta i dati per la specifica applicazione del calcolo. Le regole inizialmente così derivavano dalle proprietà (associativa, commutativa, etc.) delle operazioni, ma presto cominciano ad acquisire una loro autonomia e ad essere loro a caratterizzare le operazioni stesse. E nuove regole possono essere create per definire nuove 'primitive', nuove operazioni, nuove 'algebre' (al plurale!) caratterizzate da questi nuovi calcoli, che la matematica inglese comincerà a studiare nella seconda metà dell'ottocento, come i quaternioni, le matrici, il calcolo umbrale, basate su nuove regole del calcolo: Benjamin Peirce, padre del più famoso Charles Sanders Peirce, descrive le 162 diverse tavole moltiplicative per le algebre lineari associative, e la matematica inglese troverà nella combinatoria uno dei suoi settori di eccellenza.

Sono sviluppi che si intrecciano continuamente con quelli analoghi che emergono nella matematica tedesca: è una storia anfibia in cui si fondano i capitoli principali della matematica moderna, dalla teoria dei gruppi alla teoria delle matrici. Ma permane la differenza di stile tra le due tradizioni, che diventa anche più netta quando guardiamo allo sviluppo della logica, anche questa in comproprietà tra inglesi e tedeschi.

### 3. Nella logica moderna

La storia antica della connessione tra la realtà e la sua conoscenza linguistica e logica era basata sulla filiera aristotelica *cose* → *idee* → *parole(orali)* → *parole(scritte)*, con le 'idee' intese come 'immagini mentali' intersoggettive. I 'segni' erano invece argomento di filosofia naturale, soltanto eventi che permettevano di dedurre altri eventi, il fumo segno del fuoco e il suono delle trombe segno dell'attacco.

Ma i segni assumono un ruolo crescente nella logica medievale, sostituendo progressivamente le immagini come 'forme' della conoscenza, e soprattutto appare qualcosa di completamente nuovo: i 'segni algebrici', che non sono più come nell'antichità il 'qualcosa che sta per qualcos'altro' di S.Agostino, ma si caratterizzano come 'altra faccia' degli *algoritmi*:

- I. con l'algebra il segno si svuota di significato e di senso, della 'x' non conosciamo immediatamente il valore, ma neanche la natura: è l'incognita nelle equazioni, poi diventerà la coordinata generica nella geometria analitica, poi ancora la grandezza variabile nelle leggi fisiche, e infine il *Leerstelle*, il posto vuoto da riempire con 'qualcosa', nella logica di Frege.
- II. le cifre indo-arabe sono una rivoluzione non perché posizionali, né per la presenza dello zero, ma perché i segni in esse diventano gli 'ingredienti' essenziali degli algoritmi (le nostre operazioni in colonna), diversamente da quanto accadeva con l'abaco in cui si operava sui sassolini, e così le operazioni diventano gli algoritmi sintattici, e sono

loro i portatori del senso perduto dei segni algebrici: il significato del segno è negli algoritmi che lo utilizzano .

- III. Si afferma la coincidenza tra 'processo meccanico' e 'regola generale', il segno è anfibio, nel contempo materiale e ideale, e si lega alla macchina, un aspetto che appare forse per la prima volta con gli orologi meccanici, ed è quindi una tecnologia, ma relativa al ruolo dei segni: la stessa ruota dentata che appare nell'orologio apparirà nelle prime macchine calcolatrici di Pascal e Leibniz.

La manipolazione algebrica e il simbolismo algebrico saranno creazioni strettamente medievali, la prima già apparsa nella matematica islamica, il secondo nato dalla abitudine degli amanuensi europei di troncare, legare, condensare la scrittura dei termini più usati. E cambia anche il riferimento dei segni: soprattutto in Inghilterra si sottolinea come il significato diretto dei segni sia nelle cose reali più che nelle idee, il loro carattere sia convenzionale, e fondamentale il ruolo dell'interprete per dare loro un significato. La cultura inglese ha inoltre sempre ritenuto il pensiero scientifico perfettamente contiguo a quello comune, un raffinamento dell'origine sostanzialmente empirica e pragmatica del 'senso comune' in grado di collegare i mestieri artigianali alla filosofia naturale, e alla base della nascita della Royal Society e della Rivoluzione Industriale in terra inglese, diversamente dalla cultura tedesca che invece ha sempre intuito una distanza tra l'opinione comune e la scienza.

La logica algebrica inglese nasce non come logica della conoscenza matematica, come sarà in Frege, ma come risultato di una analisi empirica per introspezione del ragionamento comune e resterà una logica intesa come 'calcolo', o più precisamente come una 'interpretazione' delle regole del calcolo algebrico, inizialmente applicate al ragionamento deduttivo quotidiano e poi più in generale come forma necessaria e unica delle leggi fondamentali del pensiero: Boole (1815-1864) ricostruisce algebricamente la logica dalle operazioni della mente che 'riconosce' gli elementi dell'insieme e algebricamente ragiona operando sugli insiemi [4]. Non casualmente Boole, come anche poi altri logici inglesi (De Morgan, Jevons, C.S.Peirce), si era formato come algebrista.

Alla metà dell'ottocento era diventato chiaro come le regole dell'algebra potessero non essere quelle classiche dell'aritmetica: ad esempio la proprietà commutativa non valeva per i quaternioni o per le matrici; e analogamente per caratterizzare la logica algebrica occorreva aggiungere la legge di bivalenza  $x^2=x$ , la quale da sola ci dice che i valori della variabile possono solo essere 0 o 1, falso o vero. Declina il sillogismo aristotelico e, come nello stile di Leibniz e Locke, le proposizioni e definizioni logiche sono equazioni. Inoltre per la matematica dell'Ottocento il punto fondamentale riguarda la natura relazionale dei nuovi concetti matematici: numeri e figure erano normali attributi delle cose, mentre segni algebrici, derivate e integrali, limiti e serie, numeri reali, sono enti astratti esprimibili solo in forma relazionale. Ed è quindi necessario abbandonare una logica in cui la proposizione è del tipo sostanza+predicato, per una logica delle relazioni, che in Frege diventerà del tipo predicato+argomenti.[vedasi riquadro 3]

Da Leibniz discendeva l'idea ormai diffusa della connessione strettissima tra logica e matematica, che porterà al logicismo di Frege e Russell. Ma anche qui c'è una differenza tra lo "spirito di Cambridge" e lo "spirito di Göttingen". Il logicismo tedesco cercherà la genesi dei concetti aritmetici nella logica, mentre quello inglese unificherà i processi sintattici della inferenza logica e del calcolo aritmetico. La logica tedesca, sin da Leibniz e Kant, e fino a Frege, si caratterizza come una *dottrina della scienza*, e quasi per nulla come logica del ragionamento comune, e quindi si deve concentrare sullo studio dell'universo delle teorie, diversamente dalla tradizione empirista inglese che vede conoscenza comune e scientifica del tutto omogenee e che quindi si concentra sul funzionamento di una unica generica *mind*.

Tra gli inglesi sarà sempre usuale parlare della logica come di un 'calcolo', ed è un'abitudine viva ancor oggi. Sintomatico dell'impianto algebrico e calcolistico del logicismo di lingua inglese è il fatto che Russell e Whitehead giudicheranno lo 'stroke' di Sheffer [vedasi riquadro 4] come il più grande progresso della logica dopo i loro *Principia Mathematica*, mentre Hilbert e Ackermann lo giudicheranno una semplice curiosità [5]. E Whitehead sarà celebre soprattutto per i suoi studi di *algebra universale*, nei quali vedrà l'algebra della logica simbolica come parte dello studio dei "vari sistemi di ragionamento simbolico alleati all'algebra ordinaria", l'algebra universale intesa come "l'arte della manipolazione di segni sostitutivi in accordo con regole fisse e della deduzione da esse di proposizioni vere", con le operazioni del calcolo pensate come metodi di derivazione al fine di "evitare il ragionamento tramite la manipolazione simbolica" [6].

#### 4. Nell'Inghilterra dell'Ottocento

Alla stessa *Cambridge Analytical Society* appartiene Charles Babbage, creatore della *difference engine* e della *analytical engine* (programmata tramite schede perforate), un'idea non casualmente sorta riflettendo sulle tavole dei logaritmi, e sulla necessità di una ripetizione organizzata di calcoli aritmetici elementari per produrre correttamente i risultati di computazioni complesse, anche quelle dell'analisi tramite l'uso delle serie, escludendo del tutto "methods of trial or guess-work" tramite un calcolo organizzato con lo stesso livello di generalità delle formule algebriche. Anche i francesi avevano cercato di migliorare il calcolo delle tavole dei logaritmi, ma l'avevano fatto facendo lavorare assieme 80 persone che calcolavano e che sapevano fare solo le operazioni aritmetiche, dividendosi il lavoro sotto la supervisione di un matematico. Babbage invece applica l'automazione industriale a questa divisione del lavoro matematico, sostituisce i lavoratori con la macchina e il matematico con il programma



**Figura 3**  
*Lady A. A. Lovelace*

memorizzato. Idee simili appaiono in molti algebristi inglesi dell'ottocento: anche De Morgan nota come la manipolazione sintattica formale potrebbe essere realizzata da una macchina [7].

Lady Lovelace (1815-1852) è normalmente considerata una sorta di 'pasionaria del computer', altamente lodabile perché, sebbene donna, capiva qualcosa del calcolo automatico, e di fatto non vi è consenso sul suo reale contributo alla scienza dei computer. Io credo però che la figlia di Lord Byron, sebbene non contribuisca all'ingegneria della macchina, abbia intuito aspetti del calcolo sorprendentemente moderni: scrive nel 1842

[la Analytical Engine] potrebbe agire su altre cose oltre i numeri, qualora fossero oggetti le cui relazioni fondamentali potessero essere espresse da quelle della scienza astratta delle operazioni ... Supponiamo per esempio che le relazioni fondamentali dei suoni prodotti nella scienza dell'armonia e della composizione musicale fossero suscettibili di tale espressione e adattamenti, la macchina potrebbe comporre brani musicali elaborati e scientifici ad ogni grado di complessità ed estensione...

Il carattere distintivo della Analytical Engine, ... è l'introduzione in esso del principio ideato da Jacquard per regolare, tramite carte perforate, i più complicati disegni nella fabbricazione di materiali decorati ... Possiamo dire correttamente che la Analytical Engine tesse disegni algebrici così come il telaio di Jacquard tesse fiori e foglie ... come pratica possibilità, dell'idea di una macchina che pensa o ragiona [8].

Viene espressa l'idea di una strutturazione a tre livelli dell'attività computazionale: un livello esecutivo delle operazioni aritmetiche, meccanizzabile già dai tempi di Pascal, il livello algoritmico legato alla espressione algebrica del problema che fissava il loro ordine di esecuzione (la futura flow-chart), e la ricerca matematica per creare gli algoritmi da eseguire. La *analytical engine* meccanizzava anche il secondo livello, garantendo correttezza dei risultati ed economia di tempo: era la prima macchina calcolatrice programmata, antenata dei futuri computers, anche se alla base del computer moderno ci sarà anche il formalismo hilbertiano. C'è anche un primo sguardo sulla *machine intelligence*, nella osservazione che l'interpretazione delle formule e dei risultati esula dal funzionamento della macchina che semplicemente esegue "the conceptions of intelligence" registrate nel programma, e che anche attività artistiche come la composizione musicale potrebbero essere 'simulate' dalla macchina.

## 5. All'inizio del Novecento

L'abbrivio dell'algebra e della logica calcolistica inglese preludeva a nuovi strumenti del calcolo, ma per la nascita del *computer* serviva un autentico mutamento antropologico, che si delinea al passaggio fra i due secoli, un evento che possiamo chiamare l'"oblio della mente", caratterizzato dalla scomparsa del 'mondo mentale'. La scienza del XX secolo si costruirà sul rapporto diretto e univoco tra realtà e linguaggio, senza bisogno di considerare il 'pensiero' come intermediario tra i due: sarà in psicologia il comportamentismo, sarà in linguistica lo



**Figura 4**  
David Hilbert

strutturalismo, sarà in matematica il formalismo, sarà in fisica la scomparsa dei 'modelli meccanici', dalla fenomenologia di Mach all'interpretazione standard della meccanica quantistica, sarà in filosofia della scienza il neopositivismo. La fine della plurimillennaria funzione strategica della mente come ponte tra linguaggio e realtà, apre la strada alla simulazione del ragionamento che caratterizzerà la *Artificial Intelligence*: ridurre semplicemente il pensiero a linguaggio (logica più o meno standard o un autonomo 'mentalese') oppure alla realtà neurologica cerebrale (connessionismo, automi cellulari, reti neurali).

La logica di Frege aveva anche introdotto un'idea rivoluzionaria: il *calcolo*, in passato puro strumento pratico, e la *dimostrazione*, in passato la forma più pregiata della razionalità, si rivelavano coincidenti: la dimostrazione era diventata una manipolazione puramente sintattica, e il calcolo era riducibile ad una dimostrazione.

In questo clima Alan M. Turing riprende l'indagine sulle leggi del pensiero di Boole, affrontando un tema che la crisi del formalismo hilbertiano aveva lasciato aperto: la natura dell'*algoritmo* ed il suo ruolo nella intelligenza e nella scienza [1]. Lo stile del suo celebre *On computable numbers* è quello di Boole: una indagine introspettiva sul funzionamento della *mind*. Turing analizza il 'computer' (cioè l'"uomo che calcola") per formalizzarlo. Nell'articolo di Turing non appare mai la parola *intelligence*, e la parola *mind* appare negli '*states of mind*' il cui carattere puramente calcolistico è chiaro nel fatto che sono sostituibili con delle '*notes of instructions*', che semplicemente spiegano come organizzare il lavoro, e sono quindi introdotte solo per indicare l'analisi delle operazioni mentali di colui che calcola. E' il coronamento dello spirito di Cambridge in quanto nel formalismo della *macchina di Turing* si può formalizzare ogni possibile algoritmo sintattico, e la *macchina universale* delinea l'architettura del futuro computer fornendo la forma finale dell'idea generale del *calcolabile* - tutto ciò che si può fare manipolando segni secondo regole -: è il *calculus ratiocinator* delineato da Leibniz che si realizza nei linguaggi di programmazione universali. E Cantor alla fine del secolo aveva mostrato come con un insieme di numeri si potesse codificare qualsiasi insieme di dati comunque esprimibili in qualsivoglia alfabeto: sarà la *caratteristica universalis* della computer science, in quanto significa che ogni insieme di dati può essere rappresentato con stringhe di 0 e di 1 [3].

Scrive David Hilbert:

Questo gioco sintattico ha ... oltre il suo valore matematico un importante significato filosofico generale ... Pensare appare così parallelo al parlare e scrivere: formiamo proposizioni e le poniamo una dietro l'altra ... E in matematica in particolare ciò che noi consideriamo sono gli stessi segni concreti la cui forma, in accordo con la concezione adottata, è immediatamente chiara e riconoscibile.[9]

Questo gioco di formule è attuato in accordo a certe regole definite in cui si esprime la 'tecnica del nostro pensare'. Queste regole formano un sistema chiuso che può essere scoperto e stabilito definitivamente [10].

Questa idea di una riduzione sintattica della conoscenza scientifica richiama l'*ars combinatoria* di Leibniz, e fa di Hilbert un po' l'antenato della intelligenza artificiale.

Ma in Hilbert tutta l'intelligenza scientifica era ridotta alla manipolazione sintattica, ed il carattere sintattico dell'intelligenza non era una emulazione ma una realtà. Mentre invece Turing solo immagina il cervello come una macchina a stati finiti pur non ritenendolo tale (conosce la meccanica quantistica e quindi non crede esistano veramente macchine a stati discreti), e l'intelligenza di cui parlerà nel seguito sarà l'intelligenza comune' e non solo quella scientifica: anche quest'ultima è una differenza radicata tra lo 'spirito di Cambridge', affezionato alla contiguità tra scienza e conoscenza comune, e lo 'spirito di Göttingen', erede di Leibniz e votato ad una scienza come cosa separata dalla quotidianità.



**Figura 5**  
A. M. Turing

La differenza tra lo "spirito di Cambridge" e lo "spirito di Göttingen" non scompare nella storia successiva della computer science: la formalizzazione del concetto di 'algoritmo' per Turing sarà in una 'macchina' che simula le azioni di un 'uomo che calcola' sul suo quaderno, mentre per Gödel le 'funzioni ricorsive' avranno inizialmente la struttura di un sistema di equazioni la cui soluzione fornisce la funzione cercata, quasi un sistema di assiomi categorico, qualcosa che riappare anche oggi quando parliamo di 'tipi di dati astratti'. E sono anche due diverse 'architetture' computazionali: quella del 'soggetto esterno', il 'programma' che controlla l'ordine sequenziale dei calcoli nella flow-chart in un linguaggio di programmazione procedurale, e quella delle definizioni eterarchiche che si chiamano ricorsivamente a vicenda in un call-graph, anche in parallelo, in un linguaggio di programmazione logico-funzionale.

E la differenza riappare anche nella distinzione teorica tra una *semantica operativa* e una *denotazionale* per i programmi [vedasi riquadro 5].

Lo spirito di Göttingen aleggia oggi soprattutto nella informatica teorica, ma nel complesso tutta la storia del computer si svolgerà sotto il segno di una cultura empirista e calcolistica, ancora più pragmatica in terra nord-americana. E' questa cultura a creare il 'personal' computer, a fondare una computazione 'friendly', a ideare una presentazione iconica per l'uso del computer (come nel WIMP del McIntosh), e a esaltarne la pervasività. E' questa cultura a dare il format della computer science attuale, tecnologica, industriale, votata al consumo di massa. E decisamente senza storia.

## 6. Conclusioni

Ho chiamato il computer una 'epifania', un apparire alla superficie, come lo sbocco di un fiume carsico nascosto ai nostri occhi per centinaia di chilometri, e all'improvviso apparso in tutta la sua potenza anche in forme diverse in diverse culture, in quanto il pensiero formale può essere rintracciato nella storia dell'Uomo sino dalle prime civiltà neolitiche o anche prima, nelle tacche intagliate sulle ossa nel paleolitico per contare qualcosa, animali, uomini, o giorni. Ed è una intelligenza che nelle civiltà umane si è sviluppata ma anche differenziata, assumendo caratteri nuovi e diversi nelle varie civiltà.

Può sembrare che questo apra la porta ad una sorta di 'relativismo', per giunta nazionalista. Ma la storia della matematica ci insegna che questo è il suo modo di procedere storico: per quanto diversi potessero essere l'approccio di Leibniz e Newton nella creazione del calcolo differenziale, questo non ha impedito all'analisi matematica di trovare la sua struttura attuale, accettata in maniera sostanzialmente unitaria e universale.

E allora nella computer science c'è anche *la storia*, anzi forse ce n'è troppa per riuscire a digerirla facilmente. Ma è una fortuna perché fa della computer science una creazione umana, e difficilmente possiamo darci conto della sua struttura attuale e del suo futuro se non proviamo a cercarne le radici.

**Riquadro 1. Il pensiero formale: Sintassi e Semantica.**

Il pensiero formale si basa sulla manipolazione dei segni secondo le regole date dalla sintassi del linguaggio, tanto per descrivere la realtà (la rappresentazione sintattica) quanto per ragionare su di essa, ed i segni non hanno alcun significato in sé. L'attribuzione del significato ai segni è il ruolo della semantica. Il significato attribuito dalle regole dell'uso dei segni ai segni stessi riguarda le 'primitive': le regole di base della grammatica nei linguaggi formali, le regole e tabelline per le operazioni in aritmetica e algebra, il compilatore per le istruzioni di base dei linguaggi di programmazione. Gli altri segni hanno un significato variabile, non fissato da regole e a scelta di chi usa i segni: i segni interpretabili e terminali dei linguaggi formali, costanti e variabili in algebra, i dati e gli indici nei linguaggi di programmazione. Nella geometria euclidea sintassi e semantica erano indistinguibili, mentre sono nettamente distinte nella scienza moderna come logica/matematica da un lato e scienze che applicano la matematica interpretando i segni matematici dall'altro. Il computer è semplicemente la "macchina sintattica" in quanto il suo funzionamento è caratterizzato solo dalle regole della manipolazione dei segni.

**Riquadro 2. L'Entscheidungsproblem, ovvero "il problema della decisione" di Hilbert**

Hilbert (1862-1943) è il padre del formalismo matematico, che fiorì soprattutto nella sua università di Göttingen e che tende a proporre la matematica non come insieme di verità ma come formata da teorie, ciascuna delle quali basata su assiomi da cui si deducono i teoremi. Infatti Hilbert riconosceva che noi esseri umani non siamo 'infiniti': finita è la nostra vita, la nostra memoria, il nostro cervello. Ma la matematica ai suoi tempi trattava di insiemi e grandezze infinite. Come era possibile per un essere finito conoscere verità su cose infinite? Hilbert risponde che, se non possiamo trattare direttamente le cose infinite, possiamo però fare dimostrazioni finite su di esse. Tuttavia le cose finite le possiamo vedere, e quindi capire se certe proposizioni su di esse sono vere o false. Ma come facciamo a capire se una certa proposizione su cose infinite è dimostrabile o no? E' questo il "problema della decisione": trovare un algoritmo per scoprire se una certa proposizione è dimostrabile o no da certi assiomi. Per studiare un tale algoritmo Turing dovette formalizzare il concetto stesso di 'algoritmo' e a tal fine introdusse la 'macchine' che portano il suo nome.

**Riquadro 3. Frege (1848-1925) e la logica matematica.**

Frege è il padre della logica matematica, nella quale non c'è solo l'uso di un linguaggio formale apposito, ma piuttosto un mutamento radicale rispetto alla logica classica. La logica aristotelica non distingueva gli 'individui' dai concetti, e la scienza aristotelica trattava solo questi ultimi, e la 'proposizione' era intesa come **soggetto + predicato**, come in <Gli uomini sono mortali>, escludendo quindi le relazioni, mentre per Frege essa è della forma **predicato(I termine, II, termine, ..., n-esimo termine)**, come in <mangia (Giovanni, gli spaghetti, a tavola, col cucchiaio)>. Frege introduce inoltre connettivi (non, e, o, se ... allora) tra le proposizioni, e quantificatori (per ogni, esiste) che agiscono su variabili individuali per costruire proposizioni complesse, e infine la dimostrazione diventa una sequenza di proposizioni connesse da regole ben determinate, e quindi assume la forma di un calcolo, in linea di principio automatizzabile, preannuncio della fusione tra pensiero logico ed algoritmico da cui nascerà la computer science. Per Frege la stessa matematica altro non era che 'logica travestita', in quanto ogni verità matematica era traducibile e dimostrabile come verità logica.

**Riquadro 4. Lo 'stroke' di Sheffer.**

Nella logica delle proposizioni i connettivi più usati sono  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\Leftrightarrow$ , ma si scopre che alcuni di essi si possono eliminare sostituendoli con altri tramite espressioni equivalenti (l'equivalenza si verifica tramite le tavole di verità): ad esempio  $A \Rightarrow B$  è equivalente a  $\neg A \vee B$ . Sheffer scoprì che esistevano due altri connettivi: NAND(A,B) equivalente a  $\neg(A \wedge B)$  e NOR(A,B) equivalente a  $\neg(A \vee B)$ , ciascuno dei quali da solo, come unica primitiva, era sufficiente a esprimere tutti gli altri. Il nome 'stroke' deriva dal fatto che normalmente NOR(A,B) si scrive  $A | B$ , usando il simbolo  $|$  (lo 'stroke', la 'sbarretta').

**Riquadro 5. Semantica operazionale e denotazionale**

Un programma è un oggetto solo sintattico, e quindi si pone il problema (semantico) del suo 'significato'. In generale tale significato non è associato tanto allo specifico programma quanto all'algoritmo che il programma implementa in un particolare linguaggio. Tra le diverse soluzioni proposte due appaiono particolarmente rilevanti: la *semantica operazionale* riduce il significato dell'algoritmo a 'ciò che esso fa', in termini di istruzioni, transizioni e procedure, mentre la *semantica denotazionale* cerca di darne come significato 'ciò che esso è', cioè una descrizione dell'algoritmo come funzione matematica.

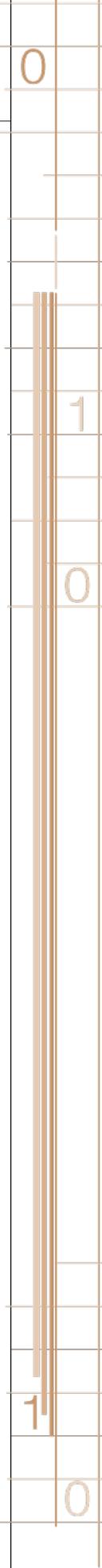
## Bibliografia

- [1] Turing, A.M. (1936-7) "On computable numbers, with an application to the Entscheidungsproblem". *Proc. of the London Math. Society*. Ser.2, 42, 230-265
- [2] Mayr, O. (1986). Authority, liberty and automatic machinery in early modern Europe. John Hopkins Univ. Press, Baltimore. (trad.ital. La bilancia e l'orologio. Il Mulino, Bologna, 1988)
- [3] Borzacchini, L. (2014) "La storia naturale dei segni: lo 'spirito di Cambridge' e 'lo spirito di Göttingen'". In: *Alan Mathison Turing: l'indecidibilità della vita*. in: C. Petrocelli (a cura di). 35-50, FrancoAngeli, Milano.
- [4] Boole, G. (1854). *An investigation of the laws of thought*. Walton and Maberly, London. (Trad. ital. Einaudi, Torino, 1976)
- [5] Burris, S. (2013). "The Algebra of Logic Tradition", in *The Stanford Encyclopedia of Philosophy* (Summer 2013 Edition), Edward N. Zalta (a cura di), <<http://plato.stanford.edu/archives/sum2013/entries/algebra-logic-tradition/>>.
- [6] Whitehead, A.N. (1898). *A Treatise on Universal Algebra*, Cambridge University Press.
- [7] De Morgan, A. (1849). *Trigonometry and double algebra*. Taylor, Walton and Maberly, London. Partially reprinted in: *From Kant to Hilbert*, William Ewald (a cura di). Clarendon Press, Oxford, 1996.
- [8] Ada Augusta, Countess of Lovelace (1842). "note A, Sketch of The Analytical Engine Invented by Charles Babbage by L. F. Menabrea with notes upon the Memoir". *Bibliothèque Universelle de Genève*, October, No. 82, <http://www.fourmilab.ch/babbage/sketch.html>
- [9] Hilbert, D. (1925). "On the infinite". Reprinted in: *From Frege to Gödel*, van Heijenoort (a cura di). Cambridge Ma., Harvard Univ. Press, 1967.
- [10] Hilbert, D. (1927). "The foundations of mathematics". Reprinted in: *From Frege to Gödel*, van Heijenoort (a cura di). Cambridge Ma., Harvard Univ. Press, 1967.

## Biografia

**Luigi Borzacchini**, docente di "Storia e Fondamenti della matematica" e di "Logica Matematica" presso il Dipartimento di Matematica dell'Università di Bari. Autore di numerosi saggi di combinatoria, di matematica discreta, di storia e filosofia della matematica, della logica e della computer science. Ha pubblicato per l'editore Dedalo "Il computer di Platone" (2005), "Il computer di Ockham" (2010) e "Il computer di Kant" (in corso di stampa), e per l'editore Garzanti l'"appendice storica" della Enciclopedia della Matematica (2013).

email: [luigi.borzacchini@uniba.it](mailto:luigi.borzacchini@uniba.it)



# Augusta Ada Lovelace (1815 - 1852)

Silvio Hénin

## Sommario

*Ada Lovelace è spesso ricordata come la prima 'programmatrice' della storia e quest'anno cade il secondo centenario della sua nascita. Anche se non mancano ottime biografie di questo personaggio, fino ad ora mai tradotte in italiano, è interessante rivisitare brevemente la sua vita, la sua opera e analizzare il suo contributo alle vicende della Macchina Analitica di Babbage. Comunque la si giudichi, Ada Lovelace resta una figura interessante, una personalità complessa dotata di grande fantasia e di intuito non comune. Una breve storia del ruolo femminile nell'informatica, soprattutto nel settore del software, chiude l'articolo.*

## Abstract

*Ada Lovelace is often remembered as the first 'programmer' of the history of computing and this year marks the second centenary of her birth. Although excellent biographies of the personage have already been published none has ever been translated into Italian, thus it seems worthy to briefly recollect her life and her work, and to scrutinize her role in the history of Babbage's Analytical Engine. In whatever way one can value Ada Lovelace's contribution, she remains an interesting figure with a complex personality, gifted with great imagination and uncommon intuition. A brief survey of the role of women in computer science, particularly in the field of software, closes the article .*

**Keywords:** Ada Lovelace, Charles Babbage, Macchina Analitica, History of automatic computing, 19th Century, Women programmers.



## 1. Nascita, giovinezza e studi di Ada

Il 10 dicembre 1815 nacque a Londra Augusta Ada Byron. Ada, come verrà sempre chiamata, fu l'unica figlia legittima del poeta romantico George Gordon, 6° barone Byron, e di Annabella Milbanke, 11<sup>a</sup> baronessa di Wentworth. La vita matrimoniale dei due genitori fu molto breve, durò infatti poco più della gravidanza di Annabella, perché i due caratteri non potevano essere peggio assortiti. Lord Byron era un poeta di grande rinomanza, uomo non bello, ma affascinante, erede di una famiglia decaduta e piena di debiti, ma anche un donnaiolo impenitente, uno scialacquatore, a volte brutale e non di rado ubriaco. Per contro, Annabella era una donna di austera razionalità e rigida religiosità, severa dominatrice, orgogliosa della sua elevata moralità, sempre tesa al totale controllo di se stessa, della sua famiglia e di tutte le persone che la circondavano e che spesso erano interessati adulatori di rango inferiore al suo. Annabella dirà di se stessa che sarebbe stata "un perfetto poliziotto o un direttore di carcere".<sup>1</sup> Poco dopo la nascita di Ada, Annabella decise di lasciare il marito e si rifugiò nella casa dei genitori, accusando Byron di violenze nei suoi confronti. Da quel momento, Lady Byron dedicherà la sua vita alla costruzione di un'immagine del poeta come quella di un uomo 'pazzo e malvagio' e dipingerà se stessa come una santa vittima. Fu probabilmente opera sua anche la diffusione di voci su un rapporto incestuoso tra Byron con la sorellastra di lui, Augusta, da cui sarebbe nata la figlia illegittima Medora.<sup>2</sup> Nell'Inghilterra dell'Ottocento era molto difficile per una donna ottenere il divorzio o la separazione, anche a fronte di conclamati vizi e tradimenti del coniuge, ancor meno probabile era mantenere l'affidamento dei figli e il patrimonio personale. Byron però non si oppose alla separazione, lasciò la tutela di Ada alla madre e partì dall'Inghilterra per il continente, per finire i suoi giorni nove anni dopo in Grecia, dove si era recato per combattere a favore dell'indipendenza di quel paese. Il poeta ricordò sempre con affetto la figlia che non ebbe più modo di vedere; le sue ultime parole furono: "Oh mia povera bambina. Mia cara Ada! Mio Dio! Avessi potuto vederla! Portatele la mia benedizione" [1-6].

Ada crebbe quindi sotto le uniche cure della madre, completamente soggetta alla sua severità ed al suo rigore. Quando Annabella era assente, cosa che succedeva spesso per le frequenti cure mediche a cui si sottoponeva,<sup>3</sup> la bambina [Fig. 1] era accudita da governanti e istitutori, accuratamente selezionati e spesso sostituiti quando non si adattavano ai ferrei concetti educativi della madre. In altri periodi Ada era affidata ad amiche della madre, che lei soprannominò 'le tre Furie', sempre sotto il controllo a distanza di

<sup>1</sup> Byron aveva soprannominato la moglie 'Principessa dei parallelogrammi' e 'Medea della matematica'.

<sup>2</sup> E' quasi certo che Medora non fosse figlia di Byron. Il poeta ebbe certamente un'altra figlia, Allegra, da Claire Claremont, cognata di Mary Shelley, l'autrice di Frankenstein. Medora fu la causa di uno lungo e squallido contrasto tra Annabella e la cognata Augusta, in cui anche ad Ada fu pesantemente coinvolta.

<sup>3</sup> Nonostante i tanti proclamati disturbi, Annabella visse fino a 68 anni, un'età ragguardevole per il suo tempo.

Annabella. L'educazione di una fanciulla aristocratica dell'Ottocento non prevedeva la frequenza di una scuola, pochi erano gli istituti che ammettevano donne e le università di Oxford e Cambridge erano totalmente precluse al sesso femminile. Perfino le biblioteche erano aperte solo ai maschi. La formazione di Ada avvenne quindi sempre nella sua casa e ogni contatto con coetanei o con persone non appartenenti al circolo virtuoso della madre le erano preclusi. Il principale obiettivo di Annabella era evitare che in Ada si manifestassero i tratti caratteriali del padre perduto, soprattutto la fantasia sfrenata e la sregolatezza. Perfino l'immagine di Byron le era nascosta: un ritratto del poeta, appeso nella casa dei nonni, era sempre coperto da un grande panno e Ada lo vide per la prima volta dopo aver compiuto i vent'anni; ogni sua curiosità al riguardo era prontamente sviata. L'educazione di Ada fu centrata sullo studio della matematica, delle scienze, della geografia e delle lingue, con limitati contatti con la letteratura fantastica e la poesia. In gioventù, la stessa Annabella era stata attratta dalla matematica, disciplina che aveva studiato assiduamente, ed era convinta che fosse un elemento insostituibile dell'educazione morale di una giovane. La piccola Ada era a volte ribelle, nonostante le severe punizioni, ma anche intelligente, curiosa e ambiziosa e si appassionò alle materie scientifiche e meccaniche. All'età di tredici anni immaginò di progettare un cavallo alato a vapore, convinta che "in circa un anno di esperimenti sarò capace di portare l'arte del volo (*Flyology*) ad una grande perfezione"[1-6].

Raggiunti i diciott'anni, Ada dovette affrontare il debutto in società, come si addiceva ad ogni giovane dell'aristocrazia. Nel 1833 fu presentata alla corte di re Guglielmo IV e, pochi giorni dopo, fece la sua comparsa al gran ballo reale.<sup>4</sup> Fondamento della vita di società per le classi nobili e i ricchi borghesi erano i ricevimenti, gli scambi di visite nei migliori salotti, i concerti, le rappresentazioni teatrali, gli scambi epistolari e, soprattutto il *gossip*, benevolo o maligno a seconda dei casi. In quei decenni era entrata in voga anche la frequentazione di conferenze, dimostrazioni scientifiche e presentazioni di nuove invenzioni che si tenevano con una certa frequenza nella capitale e la partecipazione femminile a questi eventi era perfino maggiore di quella maschile. Tra i salotti più frequentati vi era quello di Charles Babbage [Fig. 2], dove al sabato sera si riunivano fino a duecento persone, la *crème* degli intellettuali inglesi e degli

<sup>4</sup> La stagione londinese iniziava dopo Natale e si concludeva agli inizi di luglio, quando la vita a Londra diventava insopportabile per via della puzza soffocante che si diffondeva dal Tamigi, una vera cloaca a cielo aperto.



**Figura 1**  
*Augusta Ada Byron in età infantile*  
(Disegno di F. Stone).



ad Ada, che col matrimonio non diventava indipendente, ma passava semplicemente dalla tutela della madre a quella del marito. Nei successivi quattro anni Ada diede alla luce tre figli: Byron, Annabella e Ralph [1-6].

Nonostante la vita sociale, il matrimonio e le gravidanze, Ada cercò di perseverare nei suoi studi matematici e scientifici, a cui era spinta anche dall'interesse per la Macchina alle Differenze di Babbage e per la sua nuova invenzione, la Macchina Analitica [vedi Riquadro 1]. Per la matematica, sua madre l'aveva affidata a insegnanti di livello mediocre, come William King<sup>5</sup> e William Frennd; il primo era più interessato alla morale che all'algebra, il secondo era piuttosto anziano – era stato già tutore di Annabella – e antiquato: rifiutava ancora i concetti di numeri negativi e di numeri immaginari. Fu solo dopo la nascita del suo terzogenito che Ada trovò un docente migliore in Augustus De Morgan, genero di Frennd e matematico e logico di fama internazionale. Le lezioni con De Morgan si svolgevano per corrispondenza e neppure con grande frequenza. Tutti i suoi maestri riconoscevano in Ada doti di acuta intelligenza e di curiosità che li stupiva. De Morgan scrisse alla madre: “Le capacità [di Ada in matematica] sono fuori del comune per un principiante, uomo o donna. La mente di Mrs. Somerville non l'ha mai portata a null'altro che i dettagli del lavoro matematico. Lady Lovelace prenderà una strada piuttosto diversa.” [3, 4] De Morgan è inclemente verso Mary Somerville, che era ormai un'affermata autrice di libri scientifici, che leggeva Newton in latino, aveva tradotto la *Mécanique Céleste* di Laplace, e che darà alle stampe altri tre trattati.<sup>6</sup> Nei secoli XVIII e XIX, oltre alla Somerville, vi furono altre donne matematiche, ad esempio l'italiana Maria Gaetana Agnesi, che compose un testo di analisi per studenti;<sup>7</sup> la tedesca Caroline Lucretia Herschel, astronoma; l'ebrea inglese, Abigail Banich Lausada, traduttrice di Diofanto dal greco; la francese Sophie Germain che aveva studiato il teorema di Fermat. Tutte costoro erano però di estrazione borghese, mentre Ada era figlia di un barone e moglie di un conte; tra i vincoli del rango nobiliare, e questo valeva anche per i maschi, vi era l'impedimento, o almeno la disapprovazione, verso l'esercizio di una professione.<sup>8</sup> Un nobile o una nobile potevano occuparsi di letteratura, arti o scienza purché rimanessero attività dilettantesche, *hobby*, come dicono gli inglesi. Le uniche attività professionali accettabili erano, per i soli maschi, la carriera politica e quella militare. Un nobile doveva 'essere', non 'fare'.

Dopo il matrimonio, Ada fu assillata da non pochi impegni, doveva sovrintendere a tre diverse case, educare tre figli, visitare la madre quando soffriva dei suoi presunti malanni, svolgere il ruolo di moglie di un Lord, partecipando ad un

<sup>5</sup> Nessuna parentela con il marito di Ada

<sup>6</sup> *On the Connexion of the Physical Sciences* (1834), *Physical Geography* (1848), e *Molecular and Microscopic Science* (1869).

<sup>7</sup> *Istituzioni Analitiche ad uso della Gioventù Italiana* (1748).

<sup>8</sup> Per l'esattezza, l'attività di studio e di ricerca non era ancora considerata una 'professione' retribuibile, neppure per il ceto borghese; solo l'insegnamento lo era. Fu dopo la metà del secolo che cominciò la professionalizzazione della scienza. Il termine 'scienziato' (scientist) fu coniato nel 1833 da William Whewell, compagno di Babbage a Cambridge.

minimo di eventi sociali. Nonostante una nutrita servitù, il tempo che le restava per lo studio non era molto; si pensi solo ai frequenti viaggi, che potevano richiedere giorni per raggiungere da Londra la madre Annabella a Brighton o i possedimenti del marito a Horsley Towers nel Surrey e Ashley Combe nel Somerset. La sua salute non era delle migliori, nell'adolescenza una grave forma di morbillo l'aveva costretta all'immobilità per quasi tre anni e, dopo la maggiore età, era spesso soggetta ad attacchi di gastrite, asma, rinite allergica, tachicardia parossistica, disordini mestruali e problemi renali. La diagnostica dell'Ottocento attribuiva tutto ciò ad 'isteria', una mal definita patologia femminile venuta di moda in quei tempi [3]. La medicina dell'epoca era ben misera, l'arte diagnostica ancora imprecisa e la terapia disponeva di pochi mezzi: salassi, sanguisughe, purganti e sali minerali (spesso tossici, come il mercurio). Per i dolori più acuti si ricorreva al laudano, una tintura di oppio, e per contrastare la sedazione dell'oppio lo si associava a 'stimolanti', cioè gin, brandy o vino. A peggiorare il tutto si aggiungevano le cure fantasiose di ciarlatani, come il mesmerismo.<sup>9</sup> Con simili terapie le condizioni fisiche e mentali per concentrarsi nello studio non erano certo ottimali. Ada era anche distratta da altri interessi, come il pattinaggio, l'equitazione, la musica e il canto. A queste due ultime attività si dedicò intensamente per un breve periodo, fantasticando di esibirsi in pubblico – cosa impensabile e scandalosa persino per una borghese – e dichiarando che “in tre anni, sarò uguale a [Giuditta] Pasta o a [Adelaide] Kemble” [3], riferendosi alle due più applaudite cantanti liriche. Il marito e la madre la spingevano invece a concentrarsi sulla matematica, materia del cui effetto benefico, perfino sullo stato di salute, erano fermamente convinti.

### 3. La traduzione

Ada aveva raggiunto l'età di 28 anni [Fig. 3], quando venne pubblicato un articolo che descriveva sommariamente la Macchina Analitica, l'ultima invenzione di Babbage di cui Lady Lovelace non era certo all'oscuro. Quell'articolo fu il tepore primaverile che permise alle doti di Ada di fiorire, anche se solo per breve tempo. Charles Babbage, amareggiato per la brusca terminazione del suo progetto della Macchina alle Differenze, sempre più isolato dai centri decisionali, emarginato dalla potente Royal Society, avversato da scienziati influenti, come l'astronomo reale George B. Airy, aveva pensato di ottenere maggiori riconoscimenti all'estero [8-11]. Nel 1840 partì per il continente, portando con sé disegni, scritti e modelli delle sue macchine. Passando per Parigi e Ginevra, raggiunse Torino, dove era già stato vent'anni prima e dove aveva conosciuto l'astronomo Giovanni Plana, con cui era rimasto in contatto epistolare. In quell'anno si svolgeva a Torino il 2° Congresso degli Scienziati Italiani e Babbage ne approfittò per presentare ad un selezionato

<sup>9</sup> Il mesmerismo era una pratica medica inventata nel Settecento dal tedesco Franz Anton Mesmer, che consisteva nell'applicazione di calamite sulla parte del corpo che era la presunta causa della malattia. Poi si abbandonò l'uso dei magneti, nella presunzione che esistesse un 'magnetismo animale' insito nel terapeuta. La pratica era tornata di moda nell'Inghilterra dell'Ottocento. Nel mesmerismo si può trovare l'origine dell'ipnosi e della psicoanalisi.

gruppo di colleghi il suo progetto della Macchina Analitica.<sup>10</sup> Lasciando l'Italia, Babbage chiese a Plana se avrebbe potuto scrivere per lui un resoconto della presentazione da pubblicare su qualche rivista scientifica.<sup>11</sup> Plana declinò e delegò l'opera a un giovane ingegnere militare, Luigi Federico Menabrea.<sup>12</sup> [Fig. 4] Questi scrisse una breve presentazione e la pubblicò nel 1842 sulla rivista svizzera *Bibliothèque Universelle de Genève* [12]. L'articolo di Menabrea è in francese, lingua che all'epoca era usata ufficialmente nei rapporti internazionali. Quando la pubblicazione giunse in Inghilterra il fisico Charles Wheatstone, co-inventore di un telegrafo e amico di Babbage, suggerì ad Ada Lovelace di tradurla in inglese.

Ada si mise all'opera, limitandosi ad aggiungere brevi note a piè di pagina (note del traduttore). Quando Babbage lo venne a sapere e chiese ad Ada perché non avesse scritto un'opera inedita e interamente di suo pugno, Ada rispose di non averci pensato, ma si impegnò ad aggiungere note più corpose. Il lavoro fu terminato dopo alcuni mesi di frenetico lavoro, di numerosi scambi epistolari e incontri tra Ada e Babbage, di correzioni e riorcorrezioni del testo, perfino con l'aiuto di Lord Lovelace che ripassava a penna gli schemi della moglie. Le corpose Note aggiuntive, sette in tutto, allungavano di tre volte l'articolo originale di Menabrea. Ada le firmò con le sole iniziali (A. A. L.), forse non per vezzo o per modestia, ma perché non si addiceva al suo rango rendere pubblico il proprio nome. La traduzione annotata fu pubblicata sulle *Scientific Memoirs*, un periodico di recensioni e traduzioni della letteratura scientifica straniera curato da Richard Taylor [13]. Ne furono stampate solo 150 copie. Né lo scritto di Manabrea né le Note di Ada entrano nei particolari costruttivi del progetto di Babbage, ma ciò sarebbe stato impossibile; non solo gli schemi di Babbage riempivano centinaia di disegni tecnici e migliaia di pagine di appunti



**Figura 3**  
Augusta Ada Lovelace nel 1842  
(Schizzo di A. E. Chalon)

<sup>10</sup> Alcuni disegni della macchina Analitica che Babbage portò con se sono conservati presso l'Accademia delle Scienze di Torino, assieme alla corrispondenza tra Babbage e Plana, Angelo Sismonda e Federigo Sclopis.

<sup>11</sup> Babbage fu sempre restio a pubblicare lavori di suo pugno dedicati alle sue macchine matematiche. Anche per la Macchina alle Differenze aveva lasciato il compito a Dionysius Lardner. Solo nelle sue memorie, stampate nel 1864, vi dedica qualche decina di pagine, che non sono comunque una trattazione esauriente, né sistematica.

<sup>12</sup> Luigi Menabrea diventerà ministro del Regno d'Italia nei dicasteri della Marina e dei Lavori Pubblici, infine Primo Ministro dal 1867 al 1869.



**Figura 4**  
Luigi Federico Menabrea

manoscritti, per la maggior parte comprensibili solo a lui [11, 14], ma il progetto stesso era soggetto a continui ripensamenti e modifiche; si può dire che non era mai terminato. Menabrea e Ada si limitarono quindi a spiegare le funzioni della Macchina Analitica, la sua architettura generale e i suoi possibili utilizzi. L'articolo originale in francese è abbastanza lineare, anche se alcune frasi sono di non immediata interpretazione, descrive a grandi linee le componenti della macchina e mostra due esempi applicativi. Entrambi possono essere considerati 'procedure' dimostrative per la soluzione di semplici sistemi di equazioni algebriche. La prosa di Ada è più complessa di quella di Menabrea, a volte verbosa, con alcune ripetizioni e un tono leggermente enfatico, anche se la paragoniamo a quella di altri scritti femminili coevi, come i libri di Mary Somerville.<sup>13</sup>

Nella prima e lunga Nota A, Ada sottolinea le diversità tra la Macchina alle Differenze (M. D.) e la Macchina Analitica (M. A.), sostenendo che si tratta di due progetti separati ed indipendenti, mentre Babbage, nelle sue memorie, afferma esattamente il contrario, cioè che la prima è un'evoluzione naturale della seconda. Ada vuole però evitare la confusione tra le due macchine, confusione che aveva contribuito alla sospensione dei finanziamenti governativi [8-11]. Ada annuncia anche che la M. A. "lungi dall'essere più complicata [della M. D.], sarà per molti aspetti di costruzione più semplice", un'affermazione di incredibile ottimismo. La Nota A continua evidenziando il principio di separazione tra 'operazioni' e 'oggetti su cui si opera' (i dati), cioè tra unità di calcolo e memoria, dilungandosi sull'importanza della 'scienza delle operazioni', una disciplina che sarebbe, secondo Ada, trascurata dalla matematica dell'Ottocento.<sup>14</sup> In una nota a piè di pagina, Ada aggiunge che "La macchina è capace, in certe circostanze, di scoprire quale di due o più contingenze possibili si è verificata e modificare il comportamento futuro in accordo con tale evenienza", suggerendo così il concetto di 'salto condizionato', struttura imprescindibile per un calcolatore automatico, come affermerà più di un secolo dopo il teorema di

<sup>13</sup> Un paragrafo della Nota A è lungo ben 158 parole senza un punto fermo e con molte subordinate, si potrebbe dire uno stile alla James Joyce.

<sup>14</sup> Secondo la Penny Cyclopaedia, un'enciclopedia del 1834, la 'scienza delle operazioni' sarebbe l'algebra. La voce potrebbe essere stata redatta dallo stesso De Morgan. Ada trovava irragionevole che il medesimo simbolo fosse usato per una grandezza numerica e per un'operazione (ad es. il simbolo 2 in  $2x$  e in  $x^2$ ). Quello delle incongruenze delle notazioni algebriche era un tema caro a Babbage, che ne trattò più volte dal 1821 al 1830 [29].

Böhm-Jacopini.<sup>15</sup> Lady Lovelace si spinge molto oltre a Menabrea, suggerendo che “il complesso controllo [che la M. A.] ci dà sulla manipolazione esecutiva dei simboli numerici e algebrici” permetterebbe il suo uso per il calcolo simbolico, non solo numerico.<sup>16</sup> Questa visione è espressa anche dalla famosa frase “La Macchina Analitica tesse figure algebriche come il telaio di Jacquard disegna foglie e fiori”. Anche Babbage, anni prima, aveva previsto questa possibilità, ma si era reso conto che sarebbe stato necessario riprogettare la M. A. da zero: così com’era stata ideata poteva solo eseguire calcoli numerici [10].

Nelle Note B e C, Ada descrive la struttura della memoria (*store*) e delle schede perforate (*card*) usate per la programmazione [vedi Riquadro 1], sottolineando che il prelievo di un numero da un registro di memoria potrebbe avvenire in due modi: causando il suo azzeramento oppure conservando il numero stesso, alternativa a cui Menabrea non accenna. Ada stabilisce anche che le schede delle operazioni possono essere lette in una sequenza, ma anche in senso opposto (*backing*) per poter ripetere il processo. E’ un primo accenno al concetto di ‘ciclo’, anch’esso struttura fondamentale secondo Böhm-Jacopini, su cui Ada tornerà in seguito. Nel concetto di ciclo potrebbe essere sottinteso quello di ‘subroutine’, ma la traduttrice non ne parla esplicitamente. Nella Nota D si ripete l’esempio del sistema di equazioni proposto da Menabrea, ma con una notazione leggermente diversa. La successiva Nota E riprende la visione del calcolo simbolico: “La macchina può combinare le quantità numeriche esattamente come fossero lettere o qualunque altro simbolo generale, e infatti può produrre risultati in notazione algebrica, se dotata degli opportuni meccanismi [...] potrebbe sviluppare tre gruppi di risultati contemporaneamente, cioè risultati simbolici, numerici e algebrici in notazione lineare”. Nella stessa Nota, Ada ripropone il concetto di ‘ciclo’ di operazioni estendendolo alla possibilità di realizzare ‘cicli entro cicli’, quelli che oggi chiameremmo ‘*nested loops*’, principio che ella reputa utile, nella nota successiva (F), per la risoluzione di sistemi lineari di equazioni. Per chiarire gli esempi che porta, la contessa introduce una sua notazione simbolica delle operazioni, ad es. 10 (x, x, -) indica un ciclo costituito da due moltiplicazioni e una sottrazione, ripetuto 10 volte.

La Nota per cui Ada è diventata famosa è però l’ultima (Nota G), quella in cui sviluppa un programma per il calcolo dei numeri di Bernoulli<sup>17</sup>. L’esempio è molto più complesso dei due presentati da Menabrea e necessita di un ciclo di dieci operazioni che deve essere ripetuto per ogni successivo numero di Bernoulli; una variabile agisce da contatore decrementale per controllare il numero di ripetizioni [17]. Non conoscendo la struttura fisica della macchina,

<sup>15</sup> Il teorema è degli italiani Corrado Böhm e Giuseppe Jacopini [15], entrambi dello Istituto per le Applicazioni del Calcolo del CNR di Roma.

<sup>16</sup> La visione di Ada prende spunto dalla capacità della M.A. di calcolare automaticamente i segni dei risultati (+ e -) delle operazioni aritmetiche, poi la traduttrice suggerisce che si potrebbero elaborare anche i numeri immaginari e complessi.

<sup>17</sup> I numeri di Bernoulli sono una successione di numeri razionali che gioca un ruolo importante in vari problemi matematici. Bernoulli li tratta nella sua *Ars Conjectandi* del 1713 in relazione alle somme di potenze di interi successivi. I numeri di Bernoulli possono essere calcolati in diversi modi.

Ada non spiega come l'azzeramento del contatore permetterebbe di uscire dal ciclo e stampare il risultato per poi cominciare il calcolo del numero di Bernoulli successivo.<sup>18</sup> Il 'programma' è piuttosto complesso e la sua stesura ha certamente richiesto una grande concentrazione ed un continuo andirivieni di bozze e lettere tra Ada e Babbage. La Nota finale termina con una frase che cerca di smorzare ogni possibile sopravvalutazione delle reali capacità della macchina: "La Macchina Analitica non ha alcuna pretesa di originare alcunché. Essa può solo fare ciò che noi le ordiniamo di eseguire. Essa può eseguire l'analisi, ma non ha il potere di anticipare alcuna relazione o verità analitica"; la puntualizzazione si trova anche negli appunti di Babbage [14].

#### 4. Il contributo di Ada Lovelace

È difficile determinare quale fosse il contributo originale di Ada alle Note, al di là dello stile espositivo. I giudizi dei suoi biografi e degli storici dell'informatica divergono non poco, al punto di separarsi quasi in due fronti contrapposti. Da un lato troviamo le biografie di Moore e di Stein [1, 3] che sollevano qualche dubbio sulle reali capacità di Ada; dall'altro quelle di Toole e di Baum [2, 4] che tendono a dimostrare il contrario. Il primo argomento della discussione è la reale preparazione matematica di Ada, un fatto molto difficile da accertare. Le uniche fonti primarie sono le numerose lettere tra lei e i suoi insegnanti, ma molte sono andate perdute, soprattutto è introvabile un volume di appunti che Ada riempiva di esercizi e problemi durante lo studio con De Morgan. Dorothy Stein ci dice che "Incapace di assimilare i processi simbolici, solo coi quali è possibile trattare rigorosamente materie altamente complesse e astratte, Ada rimase vulnerabile ai voli di fantasia e ai sussulti mistici, alle analogie esaltanti e ai concetti pittoreschi con cui il ragionamento comune cerca di superare le difficoltà di soluzioni meccaniciste" [3]. Il biografo Benjamin Woolley aggiunge: "Il modo con cui una formula possa essere derivata da un'altra, per esempio, continuò a lasciarla perplessa e cominciò a considerare le espressioni algebriche come 'spiriti e fate', piccole creature illusorie e fastidiose che possono adottare qualunque forma desiderano" [5]. Stein fa anche notare un errore di stampa del testo di Menabrea che Ada tradusse letteralmente in modo acritico, ma il refuso era sfuggito all'autore stesso, oltre che a Babbage.

È certo che lo studio di Ada con il Dott. King e il Dott. Frennd fu, agli inizi, piuttosto povero e questo potrebbe averle lasciato lacune in algebra, in geometria analitica e in trigonometria, che si faranno sentire quando dovrà affrontare il calcolo infinitesimale e l'analisi. Anche più tardi, lo studio 'per corrispondenza' con De Morgan, irregolare e discontinuo, il poco tempo libero – a volte solo mezz'ora al giorno e non tutti i giorni – e la mancanza di vere prove d'esame, lasciano qualche perplessità. Le lodi che De Morgan, Babbage e altri rivolgevano alla competenza matematica di Ada potrebbero essere dettate da un 'effetto contrasto': su uno sfondo buio in cui le donne erano quasi totalmente

<sup>18</sup> Ada sorvola sul problema dei limiti di memoria e sul metodo di indirizzamento dei successivi numeri calcolati, ognuno dei quali dovrebbe essere memorizzato in un registro diverso. L'indirizzo del nuovo registro dovrebbe quindi essere variato ad ogni iterazione del ciclo di calcolo. Babbage aveva pensato di codificare a tale fine un 'indice' sulle schede perforate [14].

assenti e anche la maggioranza degli uomini non erano particolarmente brillanti<sup>19</sup> una stella di seconda grandezza sembrava splendere più intensamente del dovuto. Anche la differenza di rango poteva inquinare il giudizio, i suoi amici matematici erano tutti borghesi, compresa quella Mary Somerville bistrattata da De Morgan, Ada era invece una contessa, moglie di un alto funzionario reale, chi avrebbe voluto essere severo con lei? Indubbia sembra invece l'originalità del suo pensiero: Ada non si accontentava mai di imparare meccanicamente regole e procedure, ma voleva sempre capire il perché di ogni cosa. Altrettanto certe sono la sua curiosità e la sua apertura mentale. Ada immaginò che si potessero scrivere algoritmi anche per risolvere giochi come il Solitario della Bastiglia<sup>20</sup> e chiedeva a De Morgan se si sarebbe potuto estendere il piano dei numeri complessi ad una terza dimensione di numeri ipercomplessi<sup>21</sup> [3, 4].

Il secondo argomento della diatriba è la reale paternità del programma per i numeri di Bernoulli, il vero *pièce de résistance* per cui Ada è famosa. Non vi è dubbio che si tratta di un'opera impegnativa e originale, anche se la tabella del programma riprende la notazione usata da Menabrea per i suoi esempi, modificandola di poco. D'altra parte, sappiamo da Babbage stesso che "la selezione degli esempi e delle tabelle fu interamente sua [di Ada] ..., eccetto quello relativo ai numeri di Bernoulli, di cui mi sono occupato per sollevarla dalle difficoltà", anche se ammette che "ella rilevò un grossolano errore che avevo fatto" [10]. Molte delle lettere tra Ada e Babbage sono andate perdute a causa delle diatribe che scoppiarono tra l'inventore, Lord Lovelace e Annabella Byron dopo la morte di Ada, è quindi impossibile ricostruire con esattezza il reale contributo di ciascuno. Antony Hyman, biografo di Babbage, esclama drasticamente che "la leggenda di Ada come primo programmatore del mondo è priva di senso, Babbage lo fu!" [18].

Con tutto ciò, la contessa di Lovelace merita di essere ricordata, ma per qualcosa di diverso. Ada fu infatti in grado di vedere più lontano di Babbage e Menabrea, immaginando che una macchina automatica avrebbe potuto eseguire ben più dei semplici calcoli aritmetici, ma avrebbe potuto elaborare simboli che rappresentavano qualunque informazione, dall'algebra alla composizione musicale. La Nota A ci svela questa sua visione: "Il meccanismo delle operazioni può essere perfino messo in azione da qualunque oggetto su cui può operare, [...] può agire su ogni cosa, oltre ai numeri, purché tali oggetti siano assoggettabili a relazioni mutue fondamentali, esprimibili nella scienza astratta delle operazioni [...], supponiamo che le relazioni fondamentali di una nota della scienza dell'armonia e dalla composizione musicale sia suscettibile di una tale

<sup>19</sup> Nella prima metà dell'Ottocento lo stato della matematica inglese era piuttosto arretrato rispetto al continente.

<sup>20</sup> Il Solitario della Bastiglia si gioca su una scacchiera a croce con 33 posizioni di cui 32 sono occupate da pedine, lasciando libera la posizione centrale. Lo scopo è muovere una pedina alla volta, che mangia quella saltata fino a lasciarne una sola al centro.

<sup>21</sup> Questo argomento fu affrontato da William R. Hamilton proprio in quegli anni, con l'introduzione dei quaternioni.

espressione e adattamento, la macchina potrebbe comporre brani di musica scientifica di qualunque complessità e durata” [13]. La viva fantasia e l’originale acume di Ada le fecero intravedere l’immensa potenzialità di una ‘macchina universale’ per l’elaborazione simbolica dei dati, anche se impossibile da conseguire con l’architettura di Babbage. Le idee di Babbage e Ada dovranno attendere quasi cent’anni prima di essere riproposte, in forma molto diversa, quando ormai Ada Lovelace era totalmente dimenticata e di Charles Babbage si ricordava ben poco. Inoltre, è sempre prudente non cercare a tutti i costi di applicare retrospettivamente concetti moderni alle idee di un tempo così lontano, soprattutto in assenza di una trattazione esauriente, chiara e sistematica sulla Macchina Analitica da parte del suo inventore. Si rischia di ripetere la ritrita affermazione che si trattava di ‘personaggi nati prima del loro tempo’: Ada Lovelace e Charles Babbage erano, in tutto e per tutto, figli del XIX secolo [14, 20].

## 5. Gli ultimi anni di Ada

Negli ultimi giorni di lavoro sulla Traduzione, i rapporti tra Ada e Babbage si guastarono. L’inventore avrebbe voluto introdurre una prefazione anonima, un testo polemico contro il governo e le istituzioni che si rifiutavano di finanziare la costruzione delle sue macchine matematiche. Ada, giustamente, si rifiutò, pensando che un tale sfogo avrebbe fatto più male che bene ai progetti di Babbage. Si arrivò ad uno scambio epistolare in toni piuttosto accesi, soprattutto da parte di Lady Lovelace, che giunse al suo apice con un vero ultimatum: in una lunga missiva la contessa si offriva di aiutare ulteriormente Babbage, ma a condizione che lui le lasciasse il comando dell’intera operazione. Babbage, ovviamente, rifiutò [1-6]. Nonostante ciò i rapporti tornarono quelli di sempre e l’inventore rimase il benvenuto a casa Lovelace. Lord e Lady Lovelace si illudevano di potere, dall’alto dei loro titoli e grazie alle loro frequentazioni,<sup>22</sup> far pressione sul governo per continuare la costruzione delle macchine di Babbage, ma in realtà le condizioni politiche non lo permettevano: Babbage era sempre più impopolare presso i centri del potere politico e i membri della Royal Society, perfino presso il grande pubblico. L’ultimatum di Ada contiene toni di auto-esaltazione che si manifestano anche nelle lettere che in quei tempi scriveva al marito, alla madre e a vari amici. Ad esempio, Ada parla delle proprie “immense capacità di ragionamento” e della “percezione di cose che nessun altro ha, o solo pochi hanno”; arriva a declamarsi la “sacerdotessa della macchina di Babbage”, destinata a diventare “Interprete dell’Onnipotente e delle sue Leggi e suo Profeta nel mondo”, vantando lo “Immenso sviluppo della [sua] immaginazione [...], nel tempo dovuto diventerò un poeta [...] non per la mia gloria, ma per aggiungere un briciolo di sapienza al mondo”. Giunge a confrontarsi con la fama del padre: “Non credo che mio padre fosse (o avrebbe potuto essere) un Poeta più grande di quanto io sia un’Analista (e una Metafisica)” [2-4].

Il suo impegno negli studi iniziò però a diventare erratico; troppi argomenti diversi la attraevano. Incominciò ad interessarsi al sistema nervoso, all’elettricità

<sup>22</sup> L’ex -primo ministro Lord Melbourne era secondo cugino di Ada Lovelace.

animale, alle sperimentazioni elettriche di Faraday e Andrew Crosse, all'azione dei veleni. Accarezzava l'idea di una matematizzazione della fisiologia e della chimica, come era avvenuto per la fisica celeste, e auspicava l'avvento di un Newton delle scienze della vita [3]. Su suggerimento di un lontano parente che viveva in Germania, si interessò agli studi tedeschi e pensò di tradurli in inglese. Babbage, Wheatstone e Faraday le suggerirono perfino di proporsi come consigliere scientifico del principe Alberto, il consorte della regina Vittoria. Provò a scrivere recensioni letterarie e cercò di comporre poemi ad imitazione del poeta tedesco Schiller. Purtroppo si trattava di passeggeri entusiasmi destinati presto a smorzarsi. 'Fuoco fatuo', la chiamava l'amico Phillips Kay [3]. Ada ritornava a volte alla musica, ma anche in questa attività la sua dedizione non era più così costante. L'unica sua opera a stampa dopo la Traduzione furono poche note ad un libro di agronomia del marito [1, 2]. A giustificazione del suo scarso impegno, scriveva che "Il denaro è il problema, se ne avessi molto potrei educare le mie capacità per un fine" [3, 4]. Sembrerebbe una scusa improponibile da una ricca ereditiera, ma c'era un fondo di verità, come vedremo. Ada insomma si sentiva inutile, oggi diremmo 'non realizzata', ma nessun piano, nessun obiettivo sembravano più capaci di sostenerla a lungo.

Nel frattempo la contessa di Lovelace cominciò a sentire i doveri famigliari come un peso insopportabile. L'educazione dei tre figli era onerosa e lei lasciava al marito e alla madre ogni decisione in merito. Finirà che i tre bambini resteranno più affezionati alla nonna, nonostante la severa, solitaria e deprimente educazione che questa impartiva loro.<sup>23</sup> Anche i rapporti con Lord Lovelace si fecero più freddi; "io sono una 'fata', sai [...] e non desidero un marito mortale, lui [Lord Lovelace] è così terreno", scriveva Ada ad un amico [2, 3]. Cominciò così un casuale flirtare con uomini più giovani e brillanti del marito, scatenando non poche chiacchiere nei salotti londinesi. Da quasi un decennio era iniziata la sua passione per il gioco d'azzardo, le scommesse sulle corse dei cavalli, che la invischiò in un sodalizio con persone di rango inferiore e poco raccomandabili. Si è spesso parlato di un 'sistema matematico' per vincere, che sarebbe stato elaborato da Ada e Babbage, ma l'aneddoto non è verosimile: Babbage era un esperto di statistica e sapeva bene che i sistemi infallibili non esistono. Le perdite di Ada divennero sostanziose e lei dovette ricorrere a prestiti e impegnare gioielli, sempre all'insaputa di madre e marito. In verità, le risorse economiche da cui Ada poteva attingere non erano generose: 300 sterline all'anno con cui doveva pagare i suoi abiti da cerimonia, i tutori dei figli, i medici e i libri, il tutto di qualità consona al suo stato sociale [1-6]. Nell'Ottocento una donna sposata non poteva disporre neppure della propria dote o dell'eredità dei genitori, non poteva comprare o vendere beni immobili, non poteva aprire un conto bancario né chiedere ufficialmente un prestito. Tutto era affidato al marito che, graziosamente, poteva concederle una piccola somma come *argent de poche*. Nella combriccola di giocatori frequentata da Ada vi era John Crosse, il figlio di quell'Andrew Crosse che faceva esperimenti di fisica. Il rapporto della contessa Lovelace con Crosse non si limitò ad un

<sup>23</sup> Il figlio maggiore, Byron (detto Ockham), fu arruolato in marina all'età di tredici anni per correggere la sua indole ribelle.

semplice flirt, ma divenne una vera e propria relazione, che continuò anche quando Ada venne a sapere che Crosse era sposato e con figli. La relazione fu confessata alla madre e al marito negli ultimi drammatici giorni di sofferenza. Nel 1849 Ada riuscì a visitare la tenuta di Newstead Abbey nel Nottinghamshire, dimora ancestrale dei Byron fin dal 1540, sotto Enrico VIII; nel ritorno fece un lungo giro attraverso il Lake District. Fu il suo ultimo periodo di felicità [2, 3, 5].

Negli anni '40, i continui malanni di cui Ada soffriva avevano cominciato a peggiorare, in due occasioni fu soggetta a mancamenti e i dolori aumentarono. Il Dott. Locock, l'ostetrico della regina Vittoria che curava Ada, non fu in grado per lungo tempo di trovarne la causa e continuò a prescrivere i soliti rimedi palliativi: sanguisughe, laudano e 'stimolanti'. Fu solo nel 1851, grazie ad un consulto e ad una visita accurata, che ad Ada fu diagnosticato un carcinoma dell'utero.<sup>24</sup> All'epoca un intervento chirurgico era improponibile, l'anestesia era ai suoi esordi e l'asepsi ancora inesistente, neppure si disponeva di alcuna forma di chemioterapia, quindi si continuò con i soliti rimedi sintomatici. Lady Byron perseverava nella sua fiducia nel potere terapeutico del mesmerismo e voleva sottrarre Ada dalle mani dei medici per affidarla alle arti magnetiche dei guaritori. Era anche convinta del potere terapeutico dello studio e la incitava a riprendere la matematica. Gli ultimi due anni di vita furono per Ada una lunga agonia e nell'estate del 1852 ogni speranza di una guarigione era perduta. Gli ultimi suoi giorni furono resi ancor più drammatici per la confessione dei debiti di gioco e della relazione con Crosse, peccati di cui la madre incolpò non solo Lord Lovelace, per il suo scarso controllo sulla moglie, ma anche Babbage, da lei considerato un complice dei misfatti. Nell'ultimo mese, Lady Byron impedì le visite e i contatti a tutti gli amici di Ada e assunse il totale controllo della sua casa. L'ultimo a visitarla fu Charles Dickens che le leggeva i brani letterari da lei amati. Ada Lovelace si spense il 17 novembre 1852, all'età di 36 anni, la stessa a cui morì Lord Byron in Grecia. Fu sepolta vicino a Newstead Abbey, accanto alla tomba del padre. Annabella non accompagnò la salma della figlia [1-6].

Nella sua biografia, Dorothy Stein si chiede "Perché una giovane donna così promettente, con tante opportunità e aspirazioni ha realizzato così poco? " e sottolinea che casa e figli non erano in realtà un ostacolo all'impegno scientifico, marito e madre la aiutavano e la stimolavano, piuttosto sarebbero da imputare alla fragilità fisica e psichica ed alle molte malattie [3]. Forse furono anche la grande curiosità e l'accesa fantasia di Ada che ne determinarono incostanza e irresolutezza. Ada era attratta, inconsciamente, dal romanticismo del padre quanto lo era, consciamente, dal positivismo di Babbage, Faraday e Wheatstone e cercò sempre di conciliare le due tendenze in una 'scienza poetica' o una 'poesia scientifica', come traspare dalle espressioni che usa in molte sue lettere. Joan Baum riassume così la psiche di Ada : "la lava della sua immaginazione apparteneva al padre, il letto roccioso su cui scorreva era la disciplina della madre" [2]. Purtroppo la lava scorreva in modo caotico e il letto roccioso era più una costrizione che una guida sapiente.

<sup>24</sup> *Nell'Inghilterra vittoriana, per l'alone di segretezza e pudore che circondava l'intero ambito sessuale e riproduttivo, una visita ginecologica era un atto eroico che si attuava solo in casi estremi.*

Se il ricordo di Babbage e delle sue macchine rimase presente nei decenni successivi, peraltro piuttosto debolmente, quello di Ada Lovelace si ridusse a poche righe nelle numerose biografie del poeta suo padre, con velati riferimenti alla sua opera sulla Macchina Analitica. Fu solo con l'avvento dei primi calcolatori automatici e con il rinnovato interesse per Babbage che Ada tornò alla ribalta. Il ricordo della Macchina Analitica di Babbage riapparve nel 1944, durante l'inaugurazione dello Harvard Mark I, un calcolatore elettromeccanico programmabile inventato da Howard Aiken. Alla cerimonia fu invitato il canadese Richard Babbage, un pronipote di Charles, che partecipò anche al *Symposium on Large-Scale Digitale Calculating Machinery* di Harvard nel 1947, organizzato da Aiken [20]. Nel 1953 fu pubblicato uno dei primi libri sulle origini del computer, *Faster than Thought* (Più veloce del pensiero), curato dall'inglese Lord Bertram V. Bowden; nella terza pagina del volume compare un ritratto di Ada e a lei vengono dedicati alcuni paragrafi del primo capitolo [19]. Bowden fu il primo a raccogliere notizie sulla vita di Ada, attingendole direttamente dalla memoria della pronipote di lei, Lady Wentworth. Nei successivi decenni sono apparse sei diverse biografie di Ada, oltre a capitoli che le furono dedicati nei libri su Babbage e nelle storie dell'informatica [1-11]. Lentamente si costruì l'immagine e il mito di Ada come 'prima programmatrice' e 'incantatrice dei numeri', come la chiamava l'inventore inglese. Negli anni 1970 il Dipartimento della Difesa americano promosse lo sviluppo di un nuovo linguaggio di programmazione che avrebbe dovuto diventare uno standard per tutte le forze armate e che fu battezzato ADA, in memoria della giovane contessa.<sup>25</sup> Le opere di Charles Babbage e di Ada Lovelace non poterono in alcun modo contribuire agli sviluppi del computer moderno: la Macchina Analitica non fu mai costruita e il 'programma' per i numeri di Bernoulli non potè essere collaudato. Ben pochi dei pionieri dell'informatica sapevano più di qualche aneddoto sui due personaggi e il ricordo della fallita costruzione della Macchina alle Differenze agì piuttosto da deterrente su alcuni di loro.

## 5. Le eredi di Ada

La presenza femminile nel mondo del software è tutt'altro che trascurabile, soprattutto ai suoi primordi, nel periodo che va dalla II Guerra Mondiale fino agli anni '60. Già dall'inizio del conflitto l'impiego di giovani laureate in matematica per l'attività di calcolo era rilevante; la maggior parte degli uomini era stata chiamata alle armi e molte attività di supporto allo sforzo bellico dovevano essere svolte da donne. Tra gli esempi più noti di impiego di personale femminile vi è quello relativo al calcolo delle tavole di tiro per l'artiglieria americana al poligono di Aberdeen e alla Moore School di Filadelfia, lavoro che veniva svolto con il solo ausilio di calcolatrici da tavolo oppure con gli analizzatori differenziali di Vannevar Bush. La traduzione degli algoritmi di calcolo in 'programmi' e la loro esecuzione era svolta da personale femminile, civile o appartenente allo Women Army Corp. Lo stesso avveniva in Inghilterra,

<sup>25</sup> Il nome fu scelto nel 1979. Il linguaggio fu sviluppato da Jean Ichbiah alla CII Honeywell Bull. Il manuale di riferimento di ADA fu approvato il 10 dicembre 1980, anniversario di Ada, e gli fu dato il numero MIL-STD-1815 in ricordo del suo anno di nascita.

nella segretissima Government Code & Cypher School di Bletchley Park, dove si decrittavano i messaggi segreti tedeschi creati dalla macchina Enigma con le cosiddette 'Bombe', anch'esse macchine elettromeccaniche che dovevano essere 'programmate' per ogni nuovo messaggio. Nel 1944, alla GC&CS, entrò in funzione un proto-calcolatore elettronico, il Colossus, destinato alla decrittazione dei codici tedeschi 'Fish', più complessi di quelli di Enigma. L'attività era svolta dalle Women in Royal Naval Service, note col nomignolo di *wrens* (scriccioli). Anche a Los Alamos, dove si stava mettendo a punto la prima arma nucleare della storia, vi era una rilevante presenza di personale femminile che operava con calcolatrici o con macchine a schede perforate. Tra di esse troviamo Klara Dan, moglie del matematico John von Neumann, e Gertrude Blanch, che aveva già lavorato alla compilazione di tavole matematiche [21, 22].

Un caso esemplare è quello di Grace Murray Hopper, che conseguì il dottorato in matematica nel 1934 e si arruolò nella Marina degli Stati Uniti nel 1943, nel corpo delle WAVES (Women Accepted for Volunteer Emergency Service). L'anno dopo Grace Hopper fu assegnata al Bureau of Ships Computation Project dell'Università di Harvard dove lavorò come collaboratrice di Howard Aiken alla programmazione dello Harvard Mark I. Qui Hopper non solo preparò i programmi e introdusse il concetto di 'subroutine', oltre a tenere corsi di addestramento per gli operatori, ma contribuì con Aiken alla stesura del manuale di programmazione della macchina, il primo trattato di questo genere. Dopo la guerra, Hopper entrò nello staff della Eckert-Mauchly Computer Corporation, la *start-up* fondata da Presper Eckert e John Mauchly<sup>26</sup> per la produzione di calcolatori elettronici per uso civile che costruì lo UNIVAC I. Per questa macchina Hopper sviluppò uno dei primi compilatori per linguaggi di alto livello, lo A-0 del 1952, a cui fece seguire il Math-Matic e il Flow-Matic. Grace Hopper ebbe poi un ruolo di primo piano nello sviluppo del Cobol, il primo linguaggio per applicazioni gestionali a vastissima diffusione. Ritornata in Marina, si congedò definitivamente nel 1972; tredici anni dopo fu promossa al grado di Contrammiraglio. Grazie ai risultati conseguiti, le furono conferiti molti altri riconoscimenti, tra cui il Computer Sciences 'Man of the Year' del 1969. È abbastanza ironico che ad una signora si dovesse conferire il titolo di 'uomo dell'anno': l'equivalente femminile non era neppure stato previsto [23].

Un altro gruppetto di programmatrici si era formato nel 1945 attorno al primo calcolatore elettronico, lo ENIAC dell'Università di Filadelfia. Diversamente dalla Macchina Analitica e dallo Harvard Mark I, la programmazione dello ENIAC non avveniva tramite schede o nastri perforati, bensì spostando connessioni e commutatori, ma si trattava pur sempre di programmazione. Chi operava sulla macchina era un gruppo di addette: Kay Mauchley Antonelli, Jean Bartik, Betty Holberton (che lavorò anche per il Cobol), Marlyn Meltzer, Frances Spence e Ruth Teitelbaum. Con loro collaborava anche Adele Katz Goldstine, moglie di uno dei pionieri di ENIAC [24]. Nel 1948, dall'altro lato dell'Atlantico, vi era Kathleen Britten che scrisse un linguaggio assembly per lo ARC2, un computer progettato dal marito Andrew Booth al Birkbeck College [25]. L'attività

<sup>26</sup> Presper Eckert e John Mauchly erano gli inventori di ENIAC.

delle prime programmatrici non si limitava alla banale conversione degli algoritmi di calcolo in codici o istruzioni, ma richiedeva anche una certa creatività, necessaria all'ottimizzazione dei processi di calcolo; la programmatrice doveva quindi conoscere bene sia la logica delle istruzioni sia l'architettura fisica della macchina. La scarsità della memoria e la limitata velocità dei primi calcolatori elettronici imponevano spesso il ricorso a soluzioni ingegnose che le 'ragazze computer' riuscivano a ideare.

La computer science non aveva ancora, alle sue origini, un'accentuata connotazione di genere, diversamente da altre discipline scientifiche mature come la fisica, la matematica o l'ingegneria. Lentamente però le cose cambiarono e quella del programmatore divenne sempre più un'attività quasi esclusivamente maschile. I soldati tornavano dal fronte e bisognava assicurare loro un posto di lavoro, mentre molte donne tornavano ad una vita da casalinghe o all'insegnamento. I computer installati erano ancora pochi e l'uso che se ne faceva era prevalentemente il calcolo scientifico. Il lavoro di programmatore non era ancora ben definito e, nella maggioranza dei casi, i programmi erano scritti direttamente dai loro utilizzatori, matematici, fisici, ingegneri. Verso la fine degli anni '50 la situazione cominciò a cambiare, il numero di macchine installate iniziò a crescere rapidamente e i computer si diffusero in altri settori, soprattutto nelle attività gestionali e commerciali. Gli utilizzatori non erano più solo gli scienziati e le aziende che compravano calcolatori non sapevano come e dove trovare programmatori; esistevano pochi corsi di formazione e ancor meno testi didattici. Un aneddoto dà un'idea della severità del problema: un programma di formazione fu proposto ai detenuti del penitenziario di Sing Sing, chi si iscriveva e superava gli esami otteneva uno sconto di pena e trovava subito un lavoro. Ovviamente, si pensò di reclutare candidate anche nel mondo femminile. Nel 1967 la rivista americana *Cosmopolitan*, pubblicò un curioso articolo col titolo "Le ragazze del computer". Il pezzo proponeva una visione paradisiaca ed allettante della professione di programmatrice, sostenendo che vi erano già ventimila donne in quella posizione e che se ne cercavano altrettante. Il lavoro sarebbe stato ben pagato, esente da discriminazione sessista e la professione richiedeva doti considerate tipicamente femminili, come la pazienza e l'abilità di gestire i dettagli. Dopotutto, programmare un computer 'non era più difficile che preparare un pranzo' (sic). La realtà era molto meno rosea: la presenza femminile era sì piuttosto alta, circa il 22 % del totale, ma il suo ruolo era spesso subordinato, limitato alla semplice codifica, non esteso all'analisi, e la discriminazione di carriera non era del tutto assente. L'opportunità restava comunque interessante per le giovani diplomate e laureate, sostenuta anche dalla rivoluzione dei costumi e dei ruoli sociali delle donne che stava avvenendo in tutto il mondo negli anni '60. Non si sa se fu solo effetto dell'articolo di *Cosmopolitan*, ma negli anni successivi in America il numero di iscrizioni femminili ai corsi di informatica cominciò a crescere in modo rilevante. Nel 1984 la 'quota rosa' di diplomi universitari in Computer Science aveva raggiunto il 37% per il Baccellierato, il 30% per il Master e il 12% per il Dottorato – quando in ingegneria il Baccellierato femminile restava sotto il 20% e in fisica raggiungeva a stento il 30%. Poi l'andamento cominciò ad invertirsi e, nel 2006, il numero di BA femminili era ricaduto al 20%, mentre per le altre lauree

scientifiche continuava a salire. Le spiegazioni dello strano fenomeno sono molte, ma una in particolare sembra interessante: la creazione dello stereotipo del programmatore [21, 22].

Negli anni '80, dopo l'esplosione dell'informatica personale, dei PC e degli Home Computer, si creò un settore produttivo del software completamente nuovo, tanto remunerativo quanto competitivo, quello che creò giganti come Microsoft. Il settore aveva bisogno di molti addetti, disposti a lavorare assiduamente e velocemente, e li trovò non solo nel mondo dei diplomati e dei laureati, ma anche in quello di coloro che avevano abbandonato gli studi, purché abbastanza abili ed esperti per scrivere programmi. Si costituì così una sinergia tra il mondo industriale e quello degli *hacker* e dei *nerd*. Come effetto collaterale, la stampa, la televisione e il cinema coniarono e propagandarono lo stereotipo del giovane geniale programmatore: un giovane asociale, drogato dal lavoro, che riusciva a stare al terminale 20 ore al giorno e dormiva (poco) sul posto di lavoro, nutrendosi di pizza fredda e lavandosi raramente. Lo stereotipo era esattamente l'antitesi delle aspettative femminili e forse dissuase molte ragazze ad intraprendere la stessa strada. Un'altra causa fu la progressiva professionalizzazione di quella che era stata considerata un'arte, con l'introduzione di test attitudinali per la selezione che privilegiavano caratteristiche 'maschili'. Anche i corsi formali e i certificati di formazione contribuirono alla selezione, almeno negli USA. In Europa la situazione era piuttosto diversa, lo hackerismo era un fenomeno marginale e la legislazione del lavoro attenuava gli aspetti più folkloristici del lavoro del programmatore. Ciò nonostante, anche in Europa molte indagini confermano la persistenza della convinzione che l'ICT sia 'un mondo di uomini' e poche delle madri e delle educatrici intervistate ritengono che una donna potrebbe svolgere meglio di un uomo il lavoro di *system engineer* o di *software developer*. Un'indagine del 2004 nell'Europa-27 dava a meno del 25% la presenza femminile tra i laureati in informatica e meno del 28% tra i professionisti del settore. In Italia, rilevamenti del 2010-2011 davano il rapporto femmine/maschi tra le lauree in ICT a 0,21, mentre il dato per l'insieme di tutte le facoltà scientifiche era 0,66 e per il totale delle lauree 1,51 [26].

Negli USA il numero di diplomi universitari di livello più basso continuò a diminuire, ma così non avvenne per i titoli superiori, come il Master e il PhD. In questi casi le quote femminili hanno continuato a salire lentamente raggiungendo nel 2006 il 26% e il 20% rispettivamente. Ciò sta a significare una crescente presenza femminile ai livelli più creativi della professione. Come controprova, due tra gli ultimi Premi Turing, il più ambito riconoscimento della computer science, una specie di Nobel dell'informatica, sono stati assegnati a due donne: nel 2006 a Frances E. Allen e nel 2008 a Barbara Liskov. La prima per i suoi "contributi pionieristici alla teoria ed alla pratica di ottimizzazione dei compilatori". La seconda per i "contributi alla fondazione teorica dei linguaggi di programmazione e della progettazione di sistemi" [27]

### Riquadro 1 – Le macchine di Babbage

Charles Babbage progettò tre macchine per il calcolo automatico: due diverse versioni della Macchina alle Differenze e la Macchina Analitica. L'impegno occupò gran parte della sua vita professionale, dal 1820 al 1860, ma nessuno dei suoi progetti fu mai realizzato interamente ai suoi tempi.

La Macchina alle Differenze era un dispositivo destinato alla produzione di tavole matematiche, come quelle delle funzioni trigonometriche e dei logaritmi, strumenti essenziali per la navigazione marittima di un impero coloniale vasto come quello britannico. Babbage pensò di realizzare la meccanizzazione del calcolo delle tavole utilizzando il 'metodo delle differenze costanti', una procedura matematica che riduce la computazione di certe funzioni all'esecuzione di semplici somme e sottrazioni ripetute. Per la progettazione e la costruzione della macchina, Babbage ottenne un finanziamento governativo che gli permise di assumere personale tecnico specializzato, costruire le macchine utensili necessarie e dotarsi di un'officina meccanica all'avanguardia. L'opera iniziò nel 1821 e proseguì, con molti problemi gestionali e contrattuali, fino al 1832. In quell'anno era pronto solo un modello in scala ridotta pari a circa un decimo della versione definitiva, il prototipo che forse vide Ada Lovelace nel 1833 e che oggi è esposto al Science Museum di Londra. Un incrocio di dispute tra l'inventore, il governo di Sua Maestà e il costruttore Joseph Clement, decretarono la fine dei lavori nel 1834. Nel 1842 il governo decise di cessare ulteriori finanziamenti e il progetto si fermò definitivamente, cosa che lasciò Babbage amareggiato e astioso verso le istituzioni britanniche e i loro funzionari. Babbage riprese l'idea nel 1847 riprogettando l'intera macchina da zero e riuscendo a ridurre il numero di componenti da 25.000 a 8.000, aumentando la velocità di calcolo. Purtroppo, i sostenitori dell'inventore si erano ridotti ad uno sparuto gruppo di amici e la costruzione non iniziò neppure. I disegni della Macchina alle Differenze N.2, lasciati da Babbage, permisero allo Science Museum di Londra la costruzione dell'intero meccanismo tra il 1985 e il 2000, dimostrando che l'invenzione avrebbe potuto funzionare anche con la tecnologia disponibile nell'Ottocento. Oggi ne esistono due esemplari, quello di Londra e una copia al Computer History Museum di Mountain View, California.

A partire dal 1834, Babbage si concentrò su un altro progetto, da lui considerato un'evoluzione della Macchina alle Differenze, che avrebbe potuto svolgere automaticamente qualunque sequenza di calcoli aritmetici, guidata da un programma, e quindi risolvere numericamente qualunque equazione matematica. La nuova invenzione fu battezzata Macchina Analitica. Nella M. A. si separava la funzione di calcolo da quella della memoria, come nei moderni computer. L'esecuzione delle operazioni aritmetiche era affidata al *mill* (fabbrica) che doveva eseguire le quattro operazioni fondamentali dell'aritmetica, somma, sottrazione, moltiplicazione e divisione. Le operazioni potevano avvenire su numeri positivi o negativi. La memoria (*store*, magazzino) era costituita da colonne di ruote a dieci posizioni (0-9) che registravano risultati parziali e finali delle operazioni, oltre a eventuali costanti numeriche necessarie al calcolo, ad esempio il valore approssimato di  $\pi$  o il logaritmo di un numero. Ogni colonna, oggi potremmo dire ogni registro, avrebbe memorizzato un numero decimale di 50 cifre, più il segno. Babbage prevedeva da 50 a 1000 colonne. La sequenza delle operazioni era guidata da

schede perforate (*card*), idea che Babbage aveva preso dai telai automatici di Jacquard. Le schede sarebbero state di tre tipi: schede delle operazioni (*operation card*), che indicavano l'operazione da eseguire (+, -, x, :); schede delle variabili (*variable card*), che indicavano quali registri (quali colonne) erano usati per gli operandi e quali per i risultati; schede numeriche (*number card*), per introdurre nella macchina le costanti o risultati ottenuti in calcoli precedenti. Babbage prevede che la M. A. avrebbe potuto eseguire cicli ripetitivi di operazioni e anche cicli annidati (*nested loop*). Per il numero di ripetizioni necessarie ad un ciclo, Babbage pensava di perforare indici sulle schede numeriche o impiegare un quarto tipo di schede combinatorie (*combinatory card*). Grazie ad una discussione col matematico Ottaviano Fabrizio Mossotti, avvenuta durante la presentazione di Torino, Babbage prevede anche un meccanismo per il salto condizionato, cioè la possibilità di cambiare la successiva sequenza di operazioni, o di fermarsi, in base al risultato raggiunto al quel punto. Per l'esecuzione delle operazioni, soprattutto per moltiplicazioni e divisioni, era previsto l'impiego di un *barrel*, un cilindro su cui erano sistemati pioli che comandavano i singoli passi elementari dell'operazione. Il *barrel* era simile al cilindro di un *carillon*, una soluzione usata anche negli automi semoventi del Settecento. In termini moderni, si potrebbe dire che era una forma di microprogramma.

Tutte le parti della M. A. avrebbero dovuto essere organi esclusivamente meccanici: ingranaggi, leve, camme, ecc. Per il progetto Babbage inventò una 'notazione meccanica', un simbolismo che permetteva di tenere traccia dei movimenti di ogni singola parte e degli istanti in cui questi si sarebbero verificati. Secondo l'inventore, un'addizione di due numeri di 50 cifre avrebbe richiesto 1 secondo e una moltiplicazione 1 minuto, ma la macchina sarebbe stata parecchio più complessa della Macchina alle Differenze ed è difficile che avrebbe potuto muoversi così velocemente. Babbage lasciò 240 grandi disegni tecnici e 3400 pagine di appunti manoscritti, oggi conservati al British Museum, che sono ancora da analizzare sistematicamente. La tensione dell'inventore verso la minimizzazione dei tempi e dei meccanismi lo costringeva a continui ripensamenti e modifiche, per cui non sappiamo neppure quale sarebbe stata l'architettura definitiva. Della Macchina Analitica rimangono solo due frammenti dimostrativi fatti costruire da Henry Prevost Babbage, figlio di Charles, e custoditi allo Science Museum. Nel 2010 lo scrittore e informatico John Graham-Cunning ha lanciato una campagna di raccolta di fondi destinati allo studio dei progetti originali, nell'ipotesi di tentare, un giorno, la costruzione della Macchina Analitica [7-11, 14, 28].

## Bibliografia

- [1] Moore, D. L. (1977). *Ada Countess of Lovelace. Byron's Legitimate Daughter*, John Murray.
- [2] Baum, J. (1986). *The Calculating Passion of Ada Byron*, Archon Books.
- [3] Stein, D. (1987). *Ada. A Life and a Legacy*, MIT Press.
- [4] Toole, B. A. (1992). *Ada. The Enchantress of Numbers*, Strawberry Press.
- [5] Woolley, B. (1999). *The Bride of Science. Romance, Reason and Byron's Daughter*, McGraw-Hill.

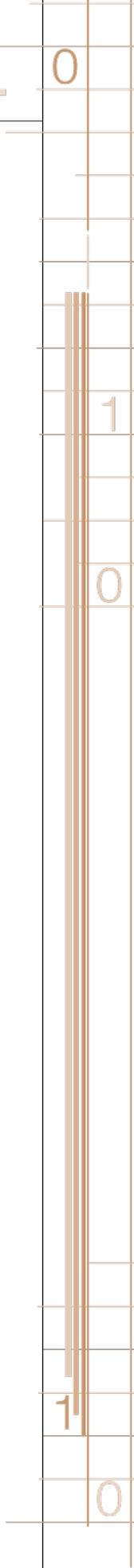
- [6] Warrick, P. S. (2007). *Charles Babbage and The Countess*, Author House.
- [7] Morrison, P., Morrison E. (1961). *Charles Babbage and his Calculating Engines*, Dover Publications.
- [8] Moseley, M. (1964). *Irascible Genius. Charles Babbage Inventor*, Hutchinson.
- [9] Hyman, A. (1982). *Charles Babbage. Pioneer of the Computer*, Princeton University Press.
- [10] Babbage, C. (1864). *Passages from the Life of a Philosopher*, Longman & Green. Ristampa Rutgers University Press, 1994.
- [11] Babbage, H. P. (1889). *Babbage's Calculating Engines*, E. & F.N. Spon, Londra. Ristampa Cambridge University Press, 2010.
- [12] Menabrea, L. F. (1842). "Notions sur la machine analytique de M. Charles Babbage", *Bibliothèque Universelle de Genève*, Nouvelle Série, vol. 41, pp. 352-276. <https://www.bibnum.education.fr/calcul-informatique/calcul/notions-sur-la-machine-analytique-de-m-charles-babbage> (ultimo accesso marzo 2015).
- [13] Menabrea, L. F. "Sketch of the Analytical Engine Invented by Charles Babbage (With notes upon the Memoir by the Translator)", *Scientific Memoirs, Selected from the Transactions of Foreign Academies of Science and Learned Societies*, vol. 3, 1843, pp. 666-731. <https://www.fourmilab.ch/babbage/sketch.html> (ultimo accesso marzo 2015).
- [14] Wilkes, M. V. (1977). "Babbage as a Computer Pioneer", *Historia Mathematica*, vol. 4, pp. 415-440.
- [15] Böhm, C., Jacopini, G. (1966). "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules", *Communications of the ACM*, 9 (5), pp. 366-371.
- [17] Wilkes, M. V. (1956), *Automatic Digital Computers*, Methuen.
- [18] Hyman, A., introduzione a Buxton, H. W. (1988). *Memoir of the Life and Labours of the Late Charles Babbage*, MIT Press.
- [19] Bowden, B. V. (1953). *Faster than Thought*, Isaac Pitman & Son.
- [20] Cohen, I.B. (1988). "Babbage and Aiken", *Ann. Hist. Comp.* 10(3), pp. 171-193.
- [21] Misa, T. J. (a cura di) (2010). *Gender Codes: Why Women Are Leaving Computing*, IEEE Computer Society.
- [22] Abbate, J. (2012). *Gender in Academic Computing: Alternative Career Paths and Norms*, MIT Press.
- [23] Beyer, K.W. (2009), *Grace Hopper and the Invention of the Information Age*, MIT Press.
- [24] Barkley Fritz, W. (1996). "The Women of ENIAC", *IEEE Ann. Hist. Comp.* 18(3), pp. 13-28.

- [25] Lavington, S. (1980). *Early British computers: the story of vintage computers and the people who built them*. Manchester University Press.
- [26] Boschetto, E. et al. (2012). *Donne e Tecnologie Informatiche*, Ed. Ca' Foscari, 2012.
- [27] Turing Award Winners. <http://amturing.acm.org/> (ultimo accesso aprile 2015).
- [28] Collier, B. (1970). *The Little Engines that Could've: The Calculating Machines of Charles Babbage*, Tesi di Dottorato, Harvard University, Cambridge, Massachusetts. <http://robroy.dyndns.info/collier/>, (ultimo accesso marzo 2015).
- [29] Dubbey, J. M. (2004). *The Mathematical Work of Charles Babbage*, Cambridge University Press.

## Biografia

**Silvio Hénin** è uno studioso di storia della tecnologia, in particolare del calcolo automatico. Consulente del Museo Nazionale di Scienza e della Tecnologia 'Leonardo da Vinci' di Milano, collabora con i periodici *Mondo Digitale*, *Le Scienze*, *IEEE Annals of the History of Computing* e ha curato numerose voci del *Dizionario Enciclopedico di Informatica, ICT e Media Digitali* edito da Enciclopedia Treccani. Autore del libro "Come le violette a primavera" (AICA, 2014), una breve storia dell'informatica. E' socio AICA (Associazione Italiana Calcolo Automatico), ACM (Association of Computer Manufacturers).

email: [silvio.henin@fastwebnet.it](mailto:silvio.henin@fastwebnet.it)



# Informatica e arte: contraddizione, rivoluzione, evoluzione

Mario Verdicchio

## Sommario

*Nello scenario piuttosto confuso delle teorie dell'arte si è affacciata nella seconda metà del XX secolo l'informatica, quando tre scienziati hanno deciso indipendentemente di usare i loro computer per creare disegni a scopo artistico. Le critiche che hanno sollevato sono le stesse critiche mosse agli albori dell'informatica, e vertono sull'impossibilità di ottenere risultati originali o sorprendenti da un computer. Il presente lavoro si propone di investigare il rapporto tra informatica e arte per dimostrare che la realtà dei fatti è significativamente diversa.*

## Abstract

*In the second half of the 20th century, computer science entered the unsettled context of theories of art thanks to three scientists, who independently started using their computers for artistic purposes. The critiques raised by such attempts are the same as the ones moved against computer science in its beginnings, and deal with the impossibility of obtaining any original or surprising results from a computer. This work aims at investigating the relation between computer science and art to show that the actual situation is significantly different.*

**Keywords:** Computation, Computer Art, Generative Art, Interactive Art, Philosophy.



## 1. Introduzione

Se l'arte è una delle attività che caratterizza l'uomo da più tempo, si pensi ad esempio alle pitture paleolitiche di animali nelle grotte di Altamira in Spagna, lo stesso non si può certamente dire dell'informatica, dal momento che il primo computer elettronico digitale risale agli anni '30, per opera di John Vincent Atanasoff nell'Iowa State College negli Stati Uniti.

Nonostante la relativa brevità della sua storia, l'informatica si è diffusa negli ultimi decenni in maniera così profonda in così tanti aspetti della nostra vita da rendere un incontro con l'arte un evento inevitabile. Tale incontro potrebbe essere interpretato in vari modi: come uno scontro tra discipline basate su concetti radicalmente diversi, come una sovrapposizione passeggera dettata dalla moda o da un desiderio di novità, come un'intersezione tra campi dello scibile umano che induce al ripensamento di alcune idee e pratiche consolidate e ne crea di nuove.

Il presente articolo si propone di gettare luce in questa direzione non per fornire delle risposte al lettore, ma per equipaggiarlo con i concetti fondamentali per aggiungersi, se lo desidera, a una delle discussioni più nuove ed effervescenti della cultura globale.

## 2. Le fondamenta incerte di informatica e arte

Per poter affrontare in maniera appropriata lo studio delle influenze reciproche di due discipline, bisognerebbe diligentemente partire dai concetti fondamentali e dalle loro definizioni per tracciare nel modo più preciso possibile i confini degli ambiti di tali discipline. Ecco che subito ci si imbatte in una questione problematica: "che cos'è un'opera d'arte?" o "che cos'è un algoritmo?" sono due domande tutt'altro che banali, e potrebbe non essere possibile fornire una risposta esauriente.

Questo non vuol dire che lo studio del rapporto tra arte e informatica sia una partita persa in partenza, ma sicuramente tale analisi richiederà una limitazione del contesto in modo da poter concentrare lo sforzo su questioni sì complesse, ma almeno affrontabili con gli strumenti a disposizione. In altre parole, si devono fare domande più specifiche di quelle sopracitate: forse non si risolveranno i nodi fondamentali alla base dell'arte e dell'informatica, ma almeno si otterrà una visione più chiara su come le due discipline si stanno vicendevolmente influenzando nelle accademie, nei laboratori e nei musei nel mondo.

### 2.1 Informatica: computazione o computer?

Il lettore più attento avrà già notato che certe restrizioni di contesto sono state già fatte nell'introduzione di questo articolo. Iniziamo dall'informatica, perché è la disciplina che pone meno problemi, almeno dal punto di vista delle definizioni fondamentali. È passato il messaggio, seppur in maniera implicita, che l'informatica ha avuto inizio con la costruzione del primo computer elettronico digitale, il che sembra implicare una definizione di questa disciplina come la "scienza dei computer elettronici digitali".

Si tratta di una definizione universalmente accettabile? A voler essere rigorosi si dovrebbe rispondere in maniera negativa perché si possono facilmente reperire esempi di risultati che rientrano a pieno titolo nell'ambito dell'informatica pur non avendo un legame diretto con tali computer: si pensi al progetto risalente alla prima metà del XIX secolo di un calcolatore interamente meccanico come lo "Analytical Engine" di Charles Babbage [1], ispirato al telaio Jacquard e programmabile per eseguire le quattro operazioni aritmetiche, oppure si pensi al "Timsort", l'algoritmo di ordinamento di numeri dalle migliori prestazioni creato da Tim Peters nel 2002 unendo le caratteristiche di due algoritmi preesistenti come il "Merge Sort" e lo "Insertion Sort" [2].

Questi due esempi mostrano chiaramente come si possa fare informatica senza computer elettronici, e quindi la definizione di cui sopra parrebbe eccessivamente restrittiva. Ovviamente è lecito chiedersi che cosa leghi i computer elettronici digitali alla macchina di Babbage o all'algoritmo di Peters, ovvero che cosa ci permette di considerare in maniera indisputabile tutti questi risultati come parte del contesto dell'informatica: questo fattore comune sarebbe il candidato più naturale per una definizione generale della disciplina.

Si tratta del fondamentale concetto di computazione, ossia dell'attività di far di conto: lo "Analytical Engine" esegue operazioni aritmetiche, il "Timsort" è una tecnica per ordinare numeri mediante una combinazione efficiente di confronti tra valori, i computer elettronici digitali sono composti da circuiti costruiti in modo tale da rispondere a degli stimoli opportuni con degli impulsi che seguono le regole della aritmetica.

Usando la lingua italiana per parlare di informatica non si apprezza questo forte, anzi, definitorio legame con la computazione, ma il discorso è ben diverso in inglese. Uno dei padri della disciplina, Alan Turing, quando parlava di "computer" in uno dei suoi più importanti lavori [3] intendeva una "persona che fa di conto", esattamente come "player" è una persona che gioca, o giocatore. Nel suo articolo, Turing presenta la sua visione di automatizzazione, per mezzo di una macchina, delle operazioni che avvengono nella testa di un essere umano mentre fa di conto. Nella seconda metà del XX secolo, quando ai pionieristici sforzi di Babbage, Atanasoff e Turing seguirono innumerevoli successi nella creazione di tali macchine, il termine "computer" andò perdendo il suo significato originale, per acquisire quello che gli associamo noi oggi, e in lingua inglese è andato consolidandosi il nome "computer science" per designare la disciplina informatica<sup>1</sup>.

Resisi conto che il nome stesso della disciplina distoglie l'attenzione dalla computazione e la focalizza sulle macchine che la realizzano, numerosi scienziati impegnati nel dibattito sullo status disciplinare dell'informatica hanno fatto diverse proposte di ridenominazione, con lo scopo di ribadire il ruolo fondamentale alla pari di altre scienze di base come la fisica e la chimica. Peter Denning, ad esempio, uno dei più quotati "opinion maker" del settore e di fatto il portavoce della Association for Computing Machinery (ACM), preferisce parlare

<sup>1</sup> In italiano è invece prevalsa la traduzione del termine "informatik", contrazione di "informazione" e "automatica", coniato dal fisico e informatico tedesco Karl Steinbuch negli anni '50 [4].

di “computing” piuttosto che di “computer science” [5], e non è un caso che il più recente e completo libro di Matti Tedre che fa il punto della situazione a riguardo sia intitolato “The Science of Computing” [6].

In realtà, almeno per quanto riguarda i nostri scopi, il dibattito concettuale su computazione e computer non è particolarmente problematico. Innanzitutto, dal punto di vista operativo qualunque forma di computazione, anche una originariamente concepita per un telaio meccanico, può essere realizzata su un computer elettronico digitale; che il cuore della disciplina risieda nel concetto più astratto di computazione e non negli artefatti concreti che sono i computer non costituisce una questione critica nello studio del rapporto tra arte e informatica perché, come vedremo presto, tale relazione è nata proprio quando degli informatici hanno iniziato, poco dopo la metà del XX secolo, a fare esperimenti con i loro computer. È un uso nuovo del computer e non una nuova applicazione della computazione che ha aperto le porte al modo di fare arte di cui vogliamo occuparci in questo articolo: se ci concentrassimo sul ruolo del calcolo puro, slegato dai computer, nell’arte, allora dovremmo anche includere in questa analisi lavori come le splendide geometrie sulle pareti dell’Alhambra a Granada oppure le meticolose tassellature di M. C. Escher, ma non è questo il nostro scopo. Invitiamo il lettore a tenere a mente la sottile ma netta distinzione tra computazione e computer, perché ci aiuterà a comprendere meglio le problematiche che affronteremo poi.

## 2.2 Arte: funzione o istituzione?

Anche se il dibattito sullo status disciplinare dell’informatica non ha ancora raggiunto una conclusione soddisfacente per tutti gli esperti del settore, almeno sui concetti fondamentali su cui tale dibattito poggia non ci sono molte controversie: ad esempio, se anche esistono diverse tecnologie con cui costruire un computer, le caratteristiche definitorie di un computer sono universalmente accettate. Tutt’altra è la situazione nel contesto dell’arte: il dibattito su che cosa sia un’opera d’arte va avanti da svariati secoli ed è stato reso ancora più acceso dalle rivoluzionarie proposte delle avanguardie del XX secolo. La varietà di modi in cui si può fare di conto, con un telaio, con carta e penna, con un circuito elettronico, ecc., non si può nemmeno paragonare alla vastità dell’orizzonte artistico umano.

L’impossibilità di fare paragoni non è una semplice iperbole: se da un lato, quello informatico, abbiamo il computer o, se vogliamo, la computazione come oggetto principale della disciplina, dal lato dell’arte non si ha alcun riferimento o, almeno, non si hanno delle caratteristiche necessarie e sufficienti universalmente accettate che ci permettano di riconoscere in maniera inequivocabile un’opera d’arte. In altre parole: se quando si fa informatica si utilizza tipicamente un computer per eseguire processi di computazione, che cosa si fa quando si fa arte?

È molto difficile rispondere a tale domanda non per una mancanza, bensì per un eccesso di risposte possibili che si possono trovare, tutte molto interessanti ma ognuna insufficiente a fornire una definizione universale di arte. Come già detto, il dibattito va avanti da tempo e non sarà certo questo articolo a risolverlo. Un

sintetico ma molto efficace compendio è fornito da Tiziana Andina [7], la quale ci aiuta a individuare i principali fattori in gioco nella creazione, nella presentazione e nella fruizione di oggetti e eventi quando questi vengono considerati opere d'arte: nel corso dei secoli diverse proposte sono state fatte, ognuna ponendo l'accento su un particolare aspetto. Il discorso inizia con Platone, secondo cui l'arte ha lo scopo di imitare la natura, a sua volta imitazione del mondo delle idee, per arrivare a teorie più recenti come quella estetica di John Dewey [8], secondo cui un'opera d'arte è tale solo se è stata concepita dall'autore con lo scopo di donare allo spettatore un'immediata quanto difficilmente analizzabile sensazione inclusiva di completezza e riesce in tale scopo, oppure la teoria artefattuale di Randall Dipert [9], secondo cui un'opera d'arte è un artefatto senza alcun risvolto pratico ma che mira solo a comunicare qualcosa di diverso da sé, o ancora la teoria istituzionale proposta da Arthur Danto [10] e perfezionata da George Dickie [11], secondo cui un'opera d'arte diviene tale nel momento in cui il suo valore viene sancito da un membro di un'istituzione non ufficiale composta da artisti, esperti e galleristi: il cosiddetto "artworld", o mondo dell'arte<sup>2</sup>.

Da notare che queste teorie sull'arte non sono state citate in ordine cronologico, a parte l'ovvia distanza temporale che ci separa da Platone, bensì in ordine decrescente rispetto alla quantità di prescrizioni sul contenuto di un'opera d'arte: partendo dall'imitazione della natura, siamo passati a definizioni sempre meno direttive, per finire con una teoria, quella istituzionale, che non pone alcuna restrizione contenutistica ma richiede solo il placet di un gruppo di persone. A voler semplificare, potremmo contrapporre la teoria istituzionale a tutte le altre, poste sotto il cappello delle teorie funzionali, ovvero teorie che attribuiscono alle opere d'arte una qualche funzione (imitativa, estetica, etc.).

In una mossa passibile di un'accusa di cinismo, potremmo adottare la teoria istituzionale e toglierci dalle spalle il peso di dover giustificare l'esistenza delle opere d'arte in funzione di un qualche scopo. Questo, però, non potrebbe essere considerato come un punto di arrivo, semplicemente perché la teoria è forse una delle proposte più problematiche dal punto di vista filosofico, dal momento che nasconde nella sua definizione una pericolosa circolarità: nega l'esistenza di criteri oggettivi che qualifichino le opere d'arte eppure richiede l'intervento di un gruppo di persone qualificate a cui delegare tale responsabilità. Le opere d'arte nascono perché scelte da persone che conoscono le opere d'arte; in altre parole, l'arte nascerebbe dal nulla.

Nonostante le sue limitazioni, crediamo che la teoria istituzionale dell'arte abbia conosciuto grande fortuna nel XX secolo perché forniva una spiegazione piuttosto semplice (semplicistica, direbbero i detrattori) a dei fenomeni nel mondo dell'arte difficilmente comprensibili sotto altri punti di vista: ci stiamo riferendo, naturalmente, alle avanguardie del primo Novecento di cui Marcel

<sup>2</sup> Non si deve scandalizzare il lettore esperto di estetica per la mancata menzione di filosofi come Baumgarten o Kant: qui non si sta parlando del "bello" (oggetto di studio dell'estetica) ma di arte. Questa distinzione, che ha portato alla nascita della filosofia dell'arte come disciplina indipendente dall'estetica, si è fatta sempre più netta nel corso del XX secolo.

Duchamp è forse il rappresentante più famoso, grazie alla sua opera "Fountain", presentata alla prima mostra della "Society of Independent Artists" a New York nel 1917, che consisteva in un urinale acquistato in un negozio di ceramiche su cui l'artista ha apposto la firma e la data "R. Mutt 1917". Un lavoro del genere difficilmente trova riscontro nelle teorie funzionali, mentre quella istituzionale lo qualifica come opera d'arte semplicemente in quanto presentata a uno degli eventi dello "artworld".

"Fountain" è solo l'esempio più noto di una lunga serie di lavori che hanno messo in discussione le fondamenta dell'arte ancor prima di poterle definire e comprendere appieno: possiamo ricordare il ferro da stiro chiodato di Man Ray ("Gift", 1921) oppure le riproduzioni di scatole di spugnette abrasive di Andy Warhol ("Brillo Box (Soap Pads)", 1964). Tuttavia, Marcel Duchamp è da considerarsi il personaggio più significativo tra tutti coloro che hanno sfidato i canoni tradizionali dell'arte di allora, almeno nel contesto di questa analisi di arte e informatica, in quanto è l'artista di riferimento di Frieder Nake, uno dei pionieri e massimi teorizzatori della cosiddetta "Computer Art".

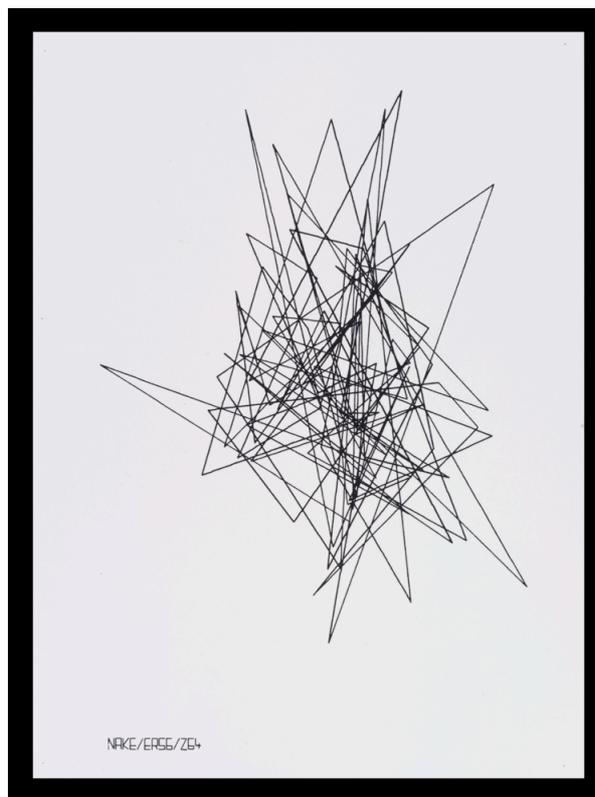
### 3. Computer Art: nuovi lavori, vecchie polemiche

Tracciare la storia degli albori della Computer Art, ossia dei primi lavori a scopo artistico realizzati con un computer, è piuttosto semplice grazie proprio a Nake, che ha sempre accompagnato alla sua attività di informatico/artista (o "algorista", come si sono definiti i primi a cimentarsi in questo campo) un significativo lavoro di cronistoria e analisi filosofica, brillantemente riassunto in un suo recente saggio [12], in cui, come accennato prima, prende Duchamp come personaggio di riferimento nell'analisi della teoria dietro l'uso dell'informatica in campo artistico. Dal momento che Nake fece ricorso al pensiero di Duchamp per difendere le prime opere di Computer Art, vediamo prima come fossero tali opere e quali accuse venissero mosse nei confronti di chi le aveva realizzate.

#### 3.1 Gli albori della Computer Art

L'inizio della Computer Art può essere fatto risalire agli anni 60 del XX secolo, quando, quasi contemporaneamente, tre informatici hanno iniziato, in maniera del tutto indipendente, a usare il computer per fare disegni geometrici: George Nees presso la Siemens a Erlangen in Germania, Michael Noll nei laboratori Bell nel New Jersey negli USA, e lo stesso Nake presso l'Università di Stoccarda, Germania. Per una curiosa coincidenza tutti i loro cognomi iniziano con la stessa lettera, ed i tre sono infatti noti con lo scherzoso nomignolo de "le tre N" della Computer Art. In realtà c'erano stati già altri esperimenti negli anni '50 che avevano previsto l'uso di computer a fini artistici, ma vogliamo considerare le tre N come i veri iniziatori della disciplina per due motivi: sono stati i primi ad usare dei computer digitali, mentre quelli usati nel decennio precedente erano sistemi analogici abbinati a degli oscilloscopi, e inoltre i loro lavori sono stati i primi ad essere esposti non nei laboratori in cui sono stati fatti gli esperimenti, bensì in vere e proprie gallerie d'arte: ad esempio, i lavori di Nake sono stati esposti, assieme ad alcuni lavori di Nees, alla galleria "Wendelin Niedlich" di Stoccarda nel novembre del 1965.

Le iniziali del cognome non sono l'unica caratteristica che accomuna i tre algoristi: i lavori da loro proposti sono tutti straordinariamente simili, a tal punto che risulta difficile credere che siano stati sviluppati in maniera del tutto indipendente. Si tratta in tutti i casi di composizioni grafiche di linee spezzate orientate in maniera casuale a formare poligoni aperti o chiusi. Nake stesso fornisce una spiegazione convincente a riguardo citando addirittura Nietzsche che avrebbe scritto nel 1882 al suo segretario Köselitz, riguardo una macchina da scrivere con sole lettere maiuscole, che "il nostro strumento di scrittura partecipa al nostro pensiero", per dire che anche una forma molto libera di pensiero come potrebbe essere la creatività di un artista sia comunque accompagnata lungo linee guida determinate dalle possibilità offerte dallo strumento prescelto per la creazione. Nel caso delle tre N tale strumento era il computer digitale degli anni '60 con le sue limitatissime capacità grafiche che includevano poco più che la funzionalità di tracciare un segmento tra due punti. Nake afferma che chiunque con una qualche velleità artistica che avesse a disposizione solo questo tipo di figura geometrica sarebbe naturalmente giunto a risultati come quello in figura 1, intitolato "Random Polygons".



**Figura 1**  
*F. Nake, "Random Polygons" (1965), stampa con plotter su carta. Dall'archivio digitale del Victoria & Albert Museum, per gentile concessione.*

Prima di analizzare il lavoro di Nake e scoprirne gli aspetti rivoluzionari, passiamo in rassegna le critiche che questo tipo di opera ha suscitato, perché tra queste vi è una polemica che ha accompagnato l'informatica fin dai suoi inizi, ancora prima che la questione dell'arte venisse sollevata, ma che sembra trovare nella Computer Art una sorta di soluzione.

### 3.2 Critiche e contraddizioni

Una critica mossa all'informatica e, in particolare, all'intelligenza artificiale, ossia la branca dell'informatica che mira a riprodurre meccanismi di ragionamento e di elaborazione delle informazioni tipicamente umani per mezzo di un computer, fu mossa ante litteram nel 1843 dalla matematica e nobildonna inglese Ada

Lovelace, quando completò la traduzione dall'italiano all'inglese del saggio del matematico Luigi Federico Menabrea sullo Analytical Engine di Babbage aggiungendo delle note personali [13]. In tali note, Lovelace dimostrava una eccezionale lungimiranza sulle possibili applicazioni future di macchine simili a quella di Babbage e aggiunse delle metodologie da lei concepite per risolvere numerosi problemi matematici, tanto da venire considerata il primo programmatore della storia: non è un caso che esista un linguaggio di programmazione chiamato "Ada", nato nel 1980 e tuttora in uso dopo numerose revisioni. Scrisse, però, anche che non bisognava aspettarsi alcuna originalità dallo Analytical Engine: esso può eseguire qualunque cosa noi siamo in grado di ordinarli di eseguire, ma non ha il potere di anticipare alcuna relazione o verità analitica.

Questa che è poi diventata nota come l'obiezione Lovelace a ogni tentativo di creare qualcosa di originale per mezzo di un computer è stata ripresa un secolo dopo da Turing nell'articolo "Computer Machinery and Intelligence" [14] in cui l'informatico propone il famoso test per valutare l'intelligenza delle macchine: anticipando critiche basate sulla suddetta obiezione alla sua visione di un futuro con macchine in grado di conversare come essere umani, Turing afferma che la matematica inglese si sarebbe ricreduta se esposta alle possibilità dell'informatica nel XX secolo.

In realtà, quello a ricredersi, se fosse stato ancora vivo, sarebbe stato lo stesso Turing, visto che, quasi un ventennio dopo, pionieri informatici come lui si sarebbero trovati ad affrontare la stessa obiezione. Per essere precisi, le critiche mosse ai lavori delle tre N erano ovviamente declinate verso un discorso di carattere artistico. Un ragionamento tipico era il seguente: dal momento che le macchine eseguono semplicemente gli ordini, non possiamo aspettarci alcuna creatività da parte loro, quindi i lavori degli algoristi, se sono frutto di un processo creativo, risultano da un processo che avviene interamente nella mente degli algoristi; gli algoristi sono matematici o ingegneri (ai tempi non erano ancora chiamati informatici) ma non artisti, quindi i loro lavori sono il frutto di un processo mentale sicuramente non artistico e perciò non possono essere considerati opere d'arte.

Il discorso si complica perché siamo di fronte alla sovrapposizione di almeno due diverse questioni: quella informatica sulla capacità di un computer di creare opere d'arte, che quindi può essere vista come una prosecuzione e una specializzazione della discussione indiretta tra Lovelace e Turing, e quella artistica sull'essenza della natura delle opere d'arte. Iniziamo dalla seconda perché è quella che presenta la contraddizione più evidente, che ci fa capire ancora meglio l'inadeguatezza della teoria istituzionale dell'arte.

In precedenza abbiamo spiegato la circolarità di questa teoria, ma le polemiche nate attorno ai lavori delle tre N mettono in luce un altro problema di carattere più pragmatico che logico: numerosi artisti respingevano i lavori di Nake e dei suoi colleghi considerandoli come giochi matematici stampati su carta, ma di fatto ci furono galleristi tedeschi e statunitensi che hanno esposto questi lavori nei loro spazi. In altre parole: come si devono considerare i lavori che suscitano reazioni contrastanti all'interno dello "artworld"? La teoria istituzionale non

0

1

0

1

0

fornisce alcuna risposta, mentre Nake fa tesoro delle parole di Duchamp: "... l'artista non è da solo quando porta a compimento l'atto creativo; c'è anche lo spettatore che stabilisce il contatto fra l'opera e il mondo esterno, decifrando e interpretando le sue qualità profonde, e che, così facendo, aggiunge il proprio contributo al processo creativo. Questo contributo è ancora più evidente nel momento in cui la posterità pronuncia il verdetto definitivo e riabilita artisti che erano dimenticati." [15]

Superando le limitazioni delle teorie preesistenti, Duchamp per la prima volta riconosce allo spettatore un ruolo comprimario nella creazione di un'opera d'arte. Se molti artisti hanno respinto sdegnati una tale proposta, Nake l'ha abbracciata completamente, e l'ha fatta sua nel rispondere alle critiche secondo cui "Random Polygons" e gli altri lavori erano stati creati "solo da dei matematici o ingegneri". I lavori delle tre N sono esteticamente molto semplici appunto perché solo matematici e ingegneri avevano accesso ai computer ed erano in grado di usarli per creare disegni. Naturalmente, se persone con una formazione artistica si fossero cimentate con la programmazione per poter realizzare i loro lavori, si sarebbero potute ottenere opere di Computer Art più complesse e interessanti. Tuttavia, ragiona Nake, se a sancire il valore di un'opera è anche il pubblico, non importa che i primi lavori di Computer Art siano stati creati da matematici piuttosto che da artisti in senso tradizionale, perché il pubblico avrebbe senza dubbio apprezzato l'innegabile contenuto rivoluzionario di questi segmenti disegnati al plotter. Una rivoluzione non solo artistica, non solo informatica, ma, come vedremo, anche culturale.

#### **4. La rivoluzione dei computer: casualità e autorialità**

Il lettore non pensi che ci siamo dimenticati della polemica a distanza temporale tra Lovelace e Turing sulla possibilità o meno di aspettarsi qualche risultato originale da un computer, specializzatasi, dopo l'avvento dei primi algoristi, nella questione se un computer possa essere caratterizzato da una qualche forma di creatività artistica. Questo problema è strettamente legato a un forte contrasto tra i principi alla base del funzionamento dei computer e quelli che invece caratterizzano l'attività creativa dell'uomo: il rigore delle regole matematiche da una parte e, diametralmente opposta dall'altra, l'assoluta libertà che sembra essere concessa agli artisti, soprattutto alla luce delle sconcertanti opere di Duchamp, Ray e Warhol.

C'è una considerazione molto semplice da fare in questo contesto: poiché i computer sono macchine per il calcolo automatico costituite da circuiti elettronici, è impossibile che possano essere creative come lo sono gli esseri umani, che sono esseri biologici coscienti con una sempre crescente esperienza del mondo alle spalle. Potrebbe suonare come una conclusione, ma in realtà è solo il nostro punto di partenza.

##### **4.1 Il compromesso sulla casualità**

I computer, dal momento che sono macchine automatiche, devono necessariamente funzionare in maniera deterministica, ossia al termine dell'esecuzione di un'operazione in un programma deve essere già

0

1

0

1

0

univocamente stabilito quale sarà l'operazione successiva da eseguire, perché, a differenza di un essere umano, un computer non può trovarsi in una situazione in cui deve scegliere come proseguire nella risoluzione di un problema. Un essere umano può prendere decisioni sulla base di esperienze passate simili a quella che sta vivendo, un computer, naturalmente, no.

Dal momento che ogni azione del computer è determinata prima di venire eseguita, il corso d'azione è interamente stabilito fin dall'inizio e le uniche variazioni che si possono avere dipendono esclusivamente dai dati in ingresso, i quali, comunque, devono rientrare negli intervalli previsti dal programma in esecuzione. Per fare un esempio dalla vita quotidiana, il computer che gestisce un bancomat può ricevere solo una gamma prestabilita di input, quella mostrata sullo schermo, e, a seconda delle scelte dell'utente umano, risponderà di conseguenza in maniera deterministica, a meno di guasti. Da questo punto di vista non vi è alcuna differenza tra il bancomat sotto casa e il più sofisticato e potente computer in un centro di calcolo della NASA. Naturalmente, nel funzionamento deterministico di un computer non c'è spazio per alcun fenomeno casuale: determinismo e casualità si escludono a vicenda. Siamo quindi di fronte a due limiti: nello svolgere le operazioni un computer non può essere creativo né agire a caso.

Il confronto tra Lovelace e Turing si è risolto indubbiamente con una netta vittoria della prima. Eppure uno dei lavori di Nake si intitola "Random Polygons", ovvero "poligoni casuali". Si tratta di un titolo ingannevole? Non esattamente. Occorre analizzare nel dettaglio le modalità con cui Nake ha concepito e realizzato la sua opera.

Dal punto di vista della creatività, intesa come quel processo mentale che ha portato alcuni informatici negli anni 60 del XX secolo a usare il loro computer per fare disegni geometrici, non possiamo che limitarci a riconoscere che le capacità grafiche degli apparati informatici di allora possono avere influenzato le scelte delle tre N. Il computer ha sì avuto un ruolo attivo, ma esclusivamente dopo che tali decisioni sono state prese: essendo una macchina deterministica non ha fatto che eseguire i comandi impartitigli dai suoi programmatori umani, che avevano deciso, appunto, di disegnare poligoni.

Dal punto di vista dell'esecuzione dell'idea, invece, l'informatica offre uno strumento molto interessante che a un'analisi superficiale potrebbe apparire come un trucco, ma che pone delle questioni epistemologiche significative: i numeri pseudocasuali. Nake non ha programmato il suo computer con istruzioni che specificassero esplicitamente le coordinate di tutti i vertici nella sua opera: tali coordinate sono state calcolate sulla base di funzioni matematiche piuttosto complesse e parametrizzate secondo diversi valori presenti all'interno del computer stesso, come ad esempio l'ora e i minuti segnati dal clock, in modo tale che, pur essendo il risultato di un calcolo perfettamente deterministico, i risultati appaiono agli occhi dell'utente umano come una sequenza di valori casuali. Il trucco è questo: un computer non è in grado di creare valori a caso ma, opportunamente programmato, può generare valori che sembrano tali.

Nake aveva un'idea abbastanza precisa del disegno che stava per creare, ma non poteva prevedere con esattezza in che posizione si sarebbero trovati i vertici dei poligoni disegnati, perché non era in grado, o semplicemente non aveva il bisogno né il desiderio, di fare i calcoli che il computer avrebbe eseguito per stabilire le coordinate di tali punti. Al momento della realizzazione del disegno, quindi, l'artista si trovò di fronte a un risultato in parte inaspettato. Turing, nel rispondere all'obiezione Lovelace, scrisse di essere convinto che i computer non fossero completamente prevedibili, ma che avessero la capacità di sorprenderci, in particolare grazie a risultati che gli esseri umani non sono in grado di ottenere immediatamente; i lavori basati sui numeri pseudocasuali sembrano dare ragione a Turing. La disciplina che fa uso dei numeri pseudocasuali dei computer per creare opere d'arte ha un nome: si chiama Generative Art, per sottolineare il carattere generativo di questo modus operandi.

Sia chiaro: esistono numerosi altri modi di ricreare l'effetto della casualità senza ricorrere all'uso di un computer. D'altronde, le funzioni matematiche che generano i numeri pseudocasuali potrebbero essere calcolate a mano, oppure si potrebbe ricorrere al fenomeno aleatorio per antonomasia: lanciare un dado. Esiste un catalogo davvero notevole di opere d'arte non al computer basate sulla casualità [16] in cui si può leggere di procedure assai curiose, come quella adottata dall'artista François Morellet nel 1971 per creare "40,000 carrés", una serie di otto serigrafie derivate da un dipinto costituito da una griglia di 200 x 200 quadrati, a ognuno dei quali l'artista associò un numero sulla base delle letture dell'elenco telefonico a cui costrinse sua moglie e i suoi figli nelle serate in famiglia; i quadrati con un numero pari vennero dipinti di blu, e quelli con un numero dispari di rosso. L'intero processo generativo durò quasi un anno.

Questo è forse un esempio estremo, ma mostra il chiaro vantaggio di lavorare con i computer: la continua evoluzione della tecnologia elettronica digitale garantisce ogni anno prestazioni migliori, ossia tempi più brevi per la realizzazione di opere anche molto complesse. Basti pensare che se nel 1965, l'anno in cui "Random Polygons" è stato creato<sup>3</sup>, un computer considerato a buon mercato per i parametri del tempo come l'IBM 1620 (prezzo: 85,000 dollari di allora) necessitava di 17,7 millisecondi per eseguire una moltiplicazione tra due numeri in virgola mobile<sup>4</sup>, oggi (gennaio 2015) con poco più di 900 dollari si può costruire un computer con un processore Intel Celeron G1830 e una scheda grafica AMD Radeon R9 295x2 (dotata anch'essa di un processore) in grado di eseguire più di 11500 miliardi di operazioni con numeri in virgola mobile al secondo.

<sup>3</sup> Per un'interessante coincidenza, il 1965 è anche l'anno di pubblicazione del famoso articolo di Gordon E. Moore, co-fondatore della Intel, in cui si prevedeva un raddoppio ogni anno del numero di transistor in un processore: la cosiddetta "legge di Moore" [17]. Le previsioni sono state successivamente corrette a un raddoppio ogni anno e mezzo.

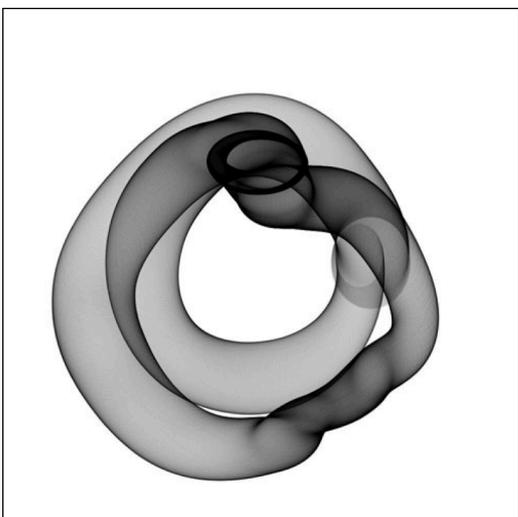
<sup>4</sup> La rappresentazione in virgola mobile di un numero razionale in un computer prevede una divisione non fissa ma, appunto, mobile tra lo spazio in memoria dedicato alla parte intera e quello dedicato alla parte decimale della mantissa del numero nella sua rappresentazione in notazione scientifica.

Per apprezzare questo sviluppo tecnologico, forse più efficace delle cifre è un lavoro contemporaneo di arte generativa, ad opera di uno dei suoi più famosi esponenti, Matt Pearson, autore anche di un libro intitolato proprio "Generative Art" [18]. L'opera si chiama "Tube Clock" ed è mostrata in figura 2. Come messo in evidenza dallo zoom al 300% di un particolare dell'opera (vedi figura 3), questa struttura tubulare è in realtà ottenuta da una miriade di ellissi a distanza molto ravvicinata l'una dall'altra. L'idea di base dell'artista di tracciare ellissi lungo un percorso circolare è accompagnata da un fattore di moderata perturbazione dato dalla pseudocasualità del rumore aggiunto ai valori delle coordinate e delle dimensioni degli ellissi. La questione delle prestazioni del computer torna alla ribalta: se anche è possibile ottenere un disegno come quello di Nike senza l'ausilio di un computer, un disegno come quello di Pearson è impossibile da realizzare con strumenti manuali, se non per un problema di tempo (pur armandosi di una pazienza come quella della famiglia Morellet), almeno per un problema di precisione e destrezza.

#### 4.2 Una nuova autorialità

La tradizionale posizione dell'artista in pieno controllo del processo di creazione dell'opera sembra essere stata minata dall'introduzione di uno strumento come il computer che, se anche del tutto privo della creatività dell'essere umano, è effettivamente dotato di caratteristiche di cui l'artista è carente: come illustrato sopra, la potenza di calcolo. Se l'introduzione della pseudocasualità fa sì che l'artista non abbia più una visione precisa di quello che sarà il risultato finale del suo lavoro, il vertiginoso aumento delle prestazioni del computer ha fatto sì che tale visione non solo si offuscasse ulteriormente (maggiore è il numero delle

operazioni pseudocasuali eseguibili dal computer nell'unità di tempo, maggiore è la variabilità del risultato finale), ma anche potesse spaziare su orizzonti altrimenti irraggiungibili. Un artista come Pearson crea opere che sono possibili solo per mezzo del computer: dobbiamo allora pensare che l'uomo abbia in qualche modo ceduto potere alla macchina? Dovrebbe Pearson condividere il merito delle sue creazioni con i computer utilizzati per realizzarle? In realtà la decisione finale e più importante è ancora saldamente nelle mani dell'artista: di tutti i possibili risultati della computazione pseudocausale, quale scegliere come opera da presentare al pubblico? Il computer riesce a esplorare in tempi ridotti lo spazio delle possibili opere, ma l'ultima parola è ancora quella

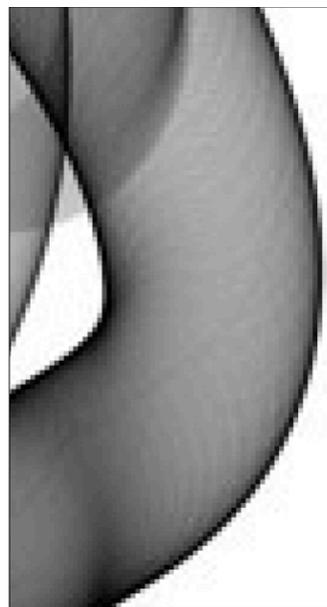


**Figura 2**  
M. Pearson, "Tube Clock" (2009),  
renderizzazione su schermo. Per gentile  
concessione dell'artista

dell'artista in quanto non esiste la possibilità di delegare tale processo di decisione al computer. Per fare ciò dovremmo essere in grado di scrivere un programma che prenda in considerazione tutti i fattori che influenzano la scelta finale dell'artista, ma già da anni filosofi come Dreyfus hanno ribadito più volte con argomenti piuttosto convincenti che ogni tentativo di costruire un elenco computabile ed esaustivo degli aspetti del pensiero umano è destinato a fallire [19], [20].

Ad ogni modo, la dipendenza di un artista dal computer per la realizzazione di un'opera d'arte non è una caratteristica determinante della Computer Art: basti pensare alla dipendenza di un pianista dal suo pianoforte, oppure di un pittore dai suoi pennelli. Da questo punto di vista, il computer è semplicemente un nuovo strumento, tecnologicamente molto avanzato, che si è aggiunto alla vasta gamma di strumenti a nostra disposizione per creare opere d'arte. La vera rivoluzione dell'autorialità che Nike aveva scoperto con i suoi lavori pionieristici non si basava su una questione di prestazioni (tra l'altro piuttosto limitate nel 1965) del computer come strumento di calcolo, bensì su una questione di astrazione, intesa come passaggio dall'esecuzione di un atto materiale alla costruzione di un modello matematico di tale atto, resa possibile dal computer inteso come macchina. Nike tiene a sottolineare la distinzione tra strumento e macchina, essendo quest'ultima un'entità ben più complessa, dotata di uno stato interno in grado di evolvere nel tempo e di tenere traccia di tale evoluzione. Per mezzo del computer, l'artista non disegna più una linea da un punto A a un punto B, e al posto dell'azione ne costruisce una descrizione sotto forma di istruzione di programma, la quale è per sua stessa natura parametrica: non fa riferimento a una singola specifica azione, ma a uno schema di cui l'azione sarà un'istanza. Come già detto prima, l'artista mantiene il controllo della creatività ma, rimuovendosi dalla dimensione delle gestualità artistiche tradizionali, la sposta da una dimensione materiale a una dimensione semiotica: il contesto di lavoro non è più fatto di pennelli e colori, bensì di segni, quelli che verranno elaborati in maniera automatica dal computer.

Il cambiamento radicale introdotto dall'uso del computer nell'arte è che l'artista non crea più un'opera, ma una classe di opere: anche se non fa ricorso ai numeri pseudocasuali un programma può essere sempre visto come istanza di un gruppo più generale di operazioni, e una semplice modifica dei suoi parametri numerici permette l'esplorazione di tale gruppo. Queste considerazioni hanno un carattere universale e trascendono dall'evoluzione della tecnologia: valevano così ai tempi di Nike come oggi, ai tempi di Pearson. Non è un caso che, quando



**Figura 3**  
Particolare di "Tube Clock", zoom al 300%.

abbiamo richiesto un'immagine ad alta risoluzione di "Tube Clock" per questo articolo, l'artista ci abbia risposto fornendoci un'altra immagine "prodotta dal sistema Tube Clock" rispetto a quella presentata sul suo sito, la cui versione "high-res" è andata persa.

## 5. Oltre Duchamp: interazione e evoluzione

Occorre fare a questo punto una precisazione per evitare che il lettore possa pensare, sulla base di quanto detto finora, che esista solo un tipo di Computer Art, ossia la Generative Art basata sui numeri pseudocasuali, e che l'evoluzione tecnologica che ci permette di avere computer con prestazioni sempre migliori giochi un ruolo sì interessante ma non significativo in questo contesto. Tali sospetti sono entrambi del tutto infondati e una sola opera di un particolare artista ci basterà a dimostrarlo.

### 5.1 I confini dell'interattività

Scott Snibbe è un artista nato a New York quattro anni dopo la presentazione di "Random Polygons", quindi chiaramente di una generazione successiva rispetto a quella delle tre N, ma anch'egli può essere a suo modo considerato un pioniere, in quanto uno dei primi a lavorare con la cosiddetta "interattività" per mezzo di un proiettore controllato da un computer. In particolare, il suo lavoro più famoso in questo senso è "Boundary Functions", presentato per la prima volta al festival "Ars Electronica" a Linz nel 1998 e poi in numerose altre mostre in giro per il mondo, di cui l'ultima nel 2008 presso il Milwaukee Art Museum negli Stati Uniti [21]. L'opera consiste nella proiezione dall'alto di linee su un pavimento in modo da dividere le persone presenti le une dalle altre (vedi figura 4).

Le linee tracciate costituiscono un diagramma di Voronoi sul pavimento, prese le teste dei partecipanti viste dall'alto come riferimenti: tale diagramma traccia i confini dei luoghi dei punti che sono più vicini a un particolare riferimento che a tutti gli altri. Il diagramma proiettato è dinamico: le linee mutano al muoversi delle persone, in modo da mantenere sempre una linea di confine tra qualunque coppia. Snibbe vuole dimostrare per mezzo di un'opera d'arte che, nonostante noi consideriamo il nostro spazio personale come appartenente interamente a noi stessi, i suoi confini sono in realtà tracciati in funzione delle persone che ci stanno intorno e cambiano spesso al di fuori del nostro controllo. L'intenzione dell'artista è quella di mostrare, in maniera quasi giocosa, l'importanza dell'accettazione delle persone intorno a noi: un messaggio ancora più carico di significato se si pensa che il titolo dell'opera è ispirato alla tesi di dottorato in matematica di Theodore Kaczynski, il serial killer noto come Unabomber.

Indipendentemente dal messaggio che Snibbe vuole trasmettere con la sua opera, è chiaro che "Boundary Functions" è un esempio di Computer Art non generativa, o meglio, non basata sull'uso di numeri pseudocasuali: il calcolo di un diagramma di Voronoi è una ben nota procedura algoritmica e, data una certa configurazione di persone sulla piattaforma, l'artista è perfettamente in grado di prevedere il risultato di tale procedura. Non ci sono sorprese e il computer non stupisce l'artista, per riprendere il tema della polemica Lovelace/Turing.



**Figura 4**

S. Snibbe, "Boundary Functions" (1998), qui presentato presso l'NTT InterCommunication Center a Tokyo, Giappone nel 1999. Immagine distribuita con licenza GFDL su [http://en.wikipedia.org/wiki/Scott\\_Snibbe](http://en.wikipedia.org/wiki/Scott_Snibbe)

Lo stupore, in realtà, è tutto del pubblico che partecipa attivamente all'opera: tale partecipazione è senza dubbio il fattore che differenzia nella maniera più significativa il lavoro di Snibbe da quelli di Nake e Pearson. Siamo di fronte a un altro tipo di Computer Art: un tipo di arte che nasce dall'interazione con il pubblico, la cosiddetta Interactive Art, o arte interattiva. Il concetto di interazione è talmente generale che occorre fare delle specificazioni. È naturalmente sempre possibile per il pubblico interagire con un'opera d'arte, anche tradizionale: di fronte a un quadro, ad esempio, un osservatore può spostarsi per osservarlo da diverse angolazioni ed ottenere così un'esperienza estetica sempre diversa; inoltre, certe opere con superfici riflettenti come il "Cloud Gate" di Anish Kapoor a Chicago (per intenderci, la gigantesca scultura di acciaio lucido a forma di fagiolo) sembrano proprio invitare il pubblico a interagire con esse, in un continuo gioco di riflessi deformati. L'interattività di un'opera di Interactive Art è però diversa perché definitoria e anzi necessaria per l'esistenza dell'opera stessa: se "Cloud Gate" può essere goduto anche a distanza, senza riflessi sulla sua superficie, con nessuna o una sola persona sulla piattaforma di "Boundary Functions" non esiste alcuna esperienza artistica, mentre con due persone sulla piattaforma si forma semplicemente una linea retta. È quando

queste due persone iniziano a muoversi o quando altre persone si aggiungono che l'opera prende vita e esprime tutto il suo potenziale.

Ricordiamoci delle parole di Duchamp riprese da Nake per difendere la Computer Art fatta da matematici e ingegneri; è facile rendersi conto del fatto che la Interactive Art riconosca al pubblico un ruolo addirittura ancora più importante di quello prescritto dall'artista di "Fountain": il pubblico non è chiamato a sancire il valore di un'opera d'arte, esso è chiamato a costruire, insieme all'artista, tale opera.

## 5.2 L'evoluzione del computer come necessità

È nel contesto della Interactive Art che diventa chiara la necessità di avere strumenti informatici dalle prestazioni adeguate. Proviamo a fare un esperimento mentale: immaginiamo di voler realizzare "Boundary Functions" senza l'aiuto dei computer. Come procederemmo? Una prima soluzione potrebbe essere quella di arricchire dal punto di vista tecnologico la piattaforma su cui si muove il pubblico con piccole bilance e LED: le bilance dovrebbero essere collegate in un reticolo tale che ogni bilancia, se premuta da un peso, trasmette parte del peso alle bilance intorno a sé; quando la situazione è stabile, i LED alle posizioni con le bilance intatte si illuminano per tracciare i confini tra le persone. Una seconda soluzione potrebbe invece sfruttare degli assistenti che, con delle torce adeguatamente modificate in modo da proiettare solo linee di luce, a seconda della configurazione del pubblico sulla piattaforma, tracciano dall'alto le linee di confine.

Naturalmente entrambe le soluzioni non solo sembrano essere di difficilissima realizzazione, ma molto probabilmente non assicurerebbero la precisione e l'estetica dei risultati che si hanno con la versione originale dell'opera.

"Boundary Functions" è resa possibile nella sua interattività dal fatto che esistono computer con un'adeguata potenza di calcolo in grado di computare in tempo reale l'output da proiettare con le linee di Voronoi sulla base delle posizioni delle persone del pubblico riconosciute tramite algoritmi di analisi dell'immagine applicati all'input proveniente dalla telecamera.

Come la macchina da scrivere di Nietzsche plasma il suo modo di pensare, così l'ispirazione di numerosi artisti viene arricchita dalle possibilità di computazione offerte dai computer: non vi è alcuna prova a riguardo, ma è ragionevole pensare che nessuno ai tempi di "Random Polygons" avrebbe potuto mai concepire un'opera come "Boundary Functions", e non perché non esistesse il concetto matematico di diagramma di Voronoi o perché non esistesse un algoritmo per risolvere il problema di tracciarne uno, bensì perché gli strumenti allora a disposizione degli artisti non avrebbero mai lasciato immaginare uno scenario in cui un computer sarebbe stato in grado di calcolare istante per istante le linee di separazione tra le persone di un gruppo in movimento su una piattaforma. Da questo punto di vista risulta più facile ammirare visionari come Turing, che più di 50 anni fa ha immaginato i computer eseguire operazioni che nemmeno oggi risultano possibili (ad esempio, conversare con una persona), nonostante i dubbi che accompagnano ogni previsione sul futuro.

## 6. Conclusione

Qualunque sia il futuro dei computer in generale e dei computer nell'arte in particolare, di fatto oggi esiste una nuova disciplina all'intersezione tra informatica e arte che è potuta nascere solo grazie all'avvento di sistemi con prestazioni adeguate a garantire interazioni in tempo reale tra persone e macchine. L'Interactive Art ha subito assunto un ruolo primario nel panorama artistico: storici dell'arte come Katja Kwastek ne hanno riconosciuto il potenziale di grande supporto alla ricerca di un'adeguata teoria dell'arte e hanno proposto una estetica dell'interazione con strumenti digitali [22], mentre filosofi dell'arte come Dominic McIver Lopes hanno addirittura fatto assurgere l'interattività a caratteristica definitoria della Computer Art in generale [23].

Nonostante le difficoltà nel riconoscere criteri universali che definiscano l'arte, l'Interactive Art, con la sua focalizzazione sulla tecnologia e sulla persona, sembra essere la disciplina che meglio incarna lo spirito del nostro tempo, con il merito di avere creato, sulle fondamenta gettate dai pionieri di metà XX secolo, un genere nuovo di opere d'arte non realizzabili con tecniche alternative perché fortemente dipendenti dalle più recenti innovazioni informatiche. La novità data dalla centralità dell'interazione tra lo spettatore e l'opera costituisce una forte rottura con il passato, forse paragonabile per portata a quella iniziata da Duchamp all'inizio del XX secolo, e considerando quello che è successo nei decenni successivi in termini di evoluzione dell'arte, della tecnologia e delle reciproche influenze, non possiamo che dirigerci con trepidazione verso ciò che ci attende nel XXI secolo.

### Riquadro 1: “Le varie declinazioni della Computer Art”

In questo articolo parliamo di Computer Art in generale per intendere una forma d'arte le cui opere sono realizzate con il fondamentale ausilio di un computer. Secondo il modo in cui è usato lo strumento informatico, tali opere possono essere classificate in maniera più specifica: abbiamo visto che se ci avvaliamo di numeri pseudocasuali si fa Generative Art, mentre se l'opera richiede l'intervento attivo dello spettatore e si trasforma dinamicamente in funzione delle sue azioni parliamo di Interactive Art.

Oltre a questi, esistono numerosi altri termini che designano la Computer Art o uno dei suoi aspetti. Ecco un elenco non esaustivo dei più comuni, da non considerarsi come mutuamente esclusivi nella classificazione di un'opera, bensì come suggerimenti per le prospettive, spesso sovrapposte, da cui osservare un'opera. Molti di questi termini, inoltre, non sono privi di problematiche di tipo concettuale.

New Media Art: forma d'arte che usa i nuovi mezzi di comunicazione. Potremmo dare per scontato che il computer sia il nuovo medium per eccellenza, ma dal punto di vista di alcuni storici dell'arte anche l'uso del televisore per installazioni artistiche rappresenta una novità, quindi non è detto che tutta la New Media Art sia necessariamente Computer Art.

Multimedia Art: forma d'arte basata sull'uso contemporaneo di mezzi di comunicazione di diverso tipo. Il computer è tradizionalmente legato al concetto di "multimediale" perché è uno strumento versatile in grado di gestire contenuti di diverso formato (testo, audio, video), ma, come nel caso precedente, la sua presenza in un'opera di Multimedia Art non può essere data per scontata.

Digital Art: disciplina artistica basata sul concetto di codifica digitale, ovvero sulla trasformazione dell'informazione da elaborare in una sequenza discreta di numeri. È sicuramente la definizione più generale, perché appunto coinvolge il principio fondamentale alla base dei computer digitali, forse fin troppo: anche una fotografia scattata con una macchina digitale potrebbe, secondo questa definizione, rientrare nell'ambito della Digital Art.

Virtual Art: l'arte che si può osservare solo per mezzo di strumenti di accesso a mondi virtuali creati con il computer. Tali strumenti sono interfacce uomo-macchina sotto forma di caschi, visori, guanti, etc. che permettono agli utenti di vedere ed eventualmente interagire con le opere d'arte presenti nell'ambiente "costruito" (le virgolette sono d'obbligo perché si tratta di una simulazione) dal computer.

Cybernetic Art: forma d'arte che mira a creare opere ispirate ai principi della cibernetica, un approccio transdisciplinare principalmente basato sulla teoria dei sistemi, che studia in particolare i sistemi dotati di un circuito di segnali che dà luogo a un feedback tra il sistema stesso e l'ambiente circostante. Tali segnali sono tipicamente codificati per mezzo di un computer, ma anche in questo caso la presenza di un computer non è essenziale.

Internet Art: nota anche come Net Art, una disciplina artistica le cui opere sfruttano le funzionalità offerte da Internet. La rete può essere semplicemente usata come canale di distribuzione alternativo al circuito tradizionale delle gallerie, oppure come parte vera e propria dell'opera d'arte, se questa si basa su servizi telematici come i siti Web o la posta elettronica, oppure se essa prevede l'interazione tra diverse persone e oggetti geograficamente distanti.

Software Art: la branca della Computer Art le cui opere sono programmi. Per riprendere la definizione di arte secondo cui un'opera è un artefatto senza un risvolto pratico, i lavori di Software Art si differenziano dagli altri programmi perché non sono da utilizzare per risolvere un problema, bensì per ottenere un'esperienza estetica.

Video Game Art: data la definizione precedente, si potrebbe pensare che questa forma d'arte abbia come oggetto i videogiochi, ma in realtà la Video Game Art è una disciplina che si li sfrutta, sia a un livello più superficiale con immagini e suoni tratti da tali giochi, sia a un livello più profondo copiando e modificandone pezzi di codice, ma per creare esperienze estetiche comunque ben distinte da una dimensione ludica.

## Bibliografia

- [1] Bromley, A. G. (1982). "Charles Babbage's Analytical Engine, 1838", *IEEE Annals of the History of Computing*, 4(3), 196-217.
- [2] Hetland, M. L. (2014). *Python Algorithms: Mastering Basic Algorithms in the Python Language*, 2<sup>nd</sup> edition, Apress.
- [3] Turing, A. (1936-7). "On computable numbers, with an application to the Entscheidungsproblem", *Proceedings of the London Mathematical Society*, 42(2), 230-265.
- [4] Steinbuch, K. (1957). "Informatik: Automatische Informationsverarbeitung", *SEL-Nachrichten*, 4, 171.
- [5] Denning, P. J. (2007). "Computing as a natural science", *Communications of the ACM*, 50(7), 13-18.
- [6] Tedre, M. (2015). *The Science of Computing*, CRC Press.
- [7] Andina, T. (2012). *Filosofie dell'arte. Da Hegel a Danto*, Carocci.
- [8] Dewey, J. (1934). *Art as Experience*, Southern Illinois University Press.
- [9] Dipert, R. R. (1993). *Artifacts, Art Works, and Agency*, Temple University Press.
- [10] Danto, A. C. (1964). "The Artworld", *Journal of Philosophy*, 61(19), 571-584.
- [11] Dickie, G. (1974). *Art and the Aesthetic: An Institutional Analysis*, Cornell University Press.
- [12] Nake, F. (2012). "Construction and Intuition: Creativity in Early Computer Art" in McCormack, J. e d'Inverno M. (a cura di) *Computers and Creativity*, Springer, 61-94.
- [13] Menabrea, L. F. (1843). "Sketch of the Analytical Engine Invented by Charles Babbage", *Scientific Memoirs, Selected from the Transactions of Foreign Academies of science and Learned Societies and from Foreign Journals*, 3.
- [14] Turing, A. (1950). "Computer Machinery and Intelligence", *Mind*, 59(236), 463-518.
- [15] Duchamp, A. (1957). "The Creative Act", intervento alla Convention of the American Federation of Arts, Houston, Texas, 3-6 aprile 1957, *Art News*, 56(4).
- [16] Malone, M. (2009). *Chance Aesthetics*, University of Chicago Press.
- [17] Moore, G. E. (1965). "Cramming More Components onto Integrated Circuits", *Electronics*, April 19, 1965, 114-117, ristampato in *Proceedings of the IEEE*, 86(1), 1998, 82-85.
- [18] Pearson, M. (2011). *Generative Art*, Manning Publications.
- [19] Dreyfus, H. (1972). *What Computers Can't Do*, MIT Press
- [20] Dreyfus, H. (1992). *What Computers Still Can't Do*, MIT Press.

[21] Snibbe, S. (1998). "Boundary Functions", [www.snibbe.com/projects/interactive/boundaryfunctions/](http://www.snibbe.com/projects/interactive/boundaryfunctions/) (ultima visita: aprile 2015)

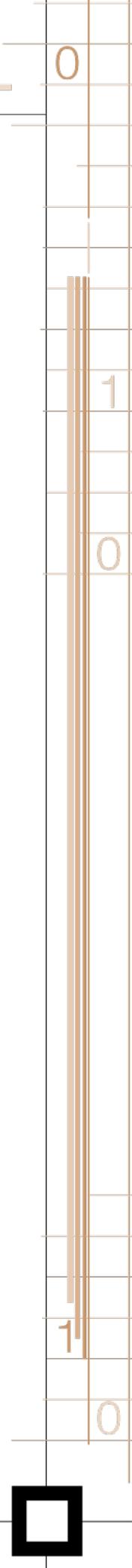
[22] Kwastek, K. (2013). *Aesthetics of Interaction in Digital Art*, MIT Press.

[23] Mclver Lopes, D. (2010). *A Philosophy of Computer Art*, Routledge.

## Biografia

**Mario Verdicchio** ha conseguito il dottorato in ingegneria dell'informazione presso il Politecnico di Milano ed è ricercatore presso l'Università degli Studi di Bergamo, dove insegna Informatica teorica presso il Dipartimento di Ingegneria e Informatica per la comunicazione presso il Dipartimento di Lingue. È cofondatore e organizzatore, insieme a colleghi della Universidade do Porto, Portogallo e della University of West Scotland, Regno Unito della serie di conferenze internazionali xCoAx: Computation, Communication, Aesthetics and X ([xcoax.org](http://xcoax.org)).

email: [mario.verdicchio@unibg.it](mailto:mario.verdicchio@unibg.it)



# Technology Intelligence: la valorizzazione delle informazioni per il miglioramento del business aziendale

Antonio Teti

## Sommario

*La Technology Intelligence è un settore scientifico che si occupa dell'identificazione delle migliori modalità di utilizzo delle tecnologie per l'ottimizzazione dell'intera infrastruttura aziendale, evidenziando le aree e i processi in cui sussistono inefficienze, debolezze, costi sprechi che possono compromettere la produttività e lo sviluppo dell'azienda. È un sistema che mira a catturare e diffondere una cultura aziendale rivolta all'analisi delle informazioni raccolte, filtrate e analizzate dalle tecnologie fruibili dall'impresa. L'obiettivo è di produrre una "saggezza" aziendale utile per individuare le migliori decisioni da assumere per ottimizzare il business dell'organizzazione.*

## Abstract

*Technology Intelligence is a field of science that deals with the identification of the best ways to use technology for the optimization of the entire corporate infrastructure, highlighting areas and processes where there are inefficiencies, weaknesses, costs wastes that can compromise productivity and the development of the company. It is a system that aims to capture and disseminate a corporate culture focused on the analysis of the information collected, filtered and analyzed by the technologies usable by the company. The goal is to produce a "wisdom" corporate useful for identifying the best decisions to be made to optimize the business organization.*

**Keywords:** Technology, Intelligence, IT Governance, Information and Communication Technology, ICT Management



## 1. Fondamenti di Technology Intelligence

Risulta evidente che le innovazioni nel settore ICT possono risultare cruciali per il raggiungimento del successo aziendale, indipendentemente dalle caratteristiche settoriali, organizzative e strutturali di ogni impresa. Individuare le costanti innovazioni del settore delle tecnologie informatiche e adoperarsi per anticiparne i possibili mutamenti che possono provocare a qualsiasi livello, risulta essenziale per i vertici decisionali di qualsiasi organizzazione. Pertanto, l'aggiornamento continuo dell'ambiente tecnologico, soprattutto in un'era di continui e repentini cambiamenti geopolitici, è una condizione essenziale per una gestione responsabile e corretta delle informazioni.

Cohen e Levinthal, in un loro saggio [1] molto apprezzato, introducono il concetto di "*absorptive capacity*" (capacità di assorbimento), che si traduce nella competenza che consente di riconoscere la *conoscenza esterna* rilevante per l'impresa, e di acquisirla ed integrarla nel proprio patrimonio di competenze esistenti. In altri termini, quanto più uno scenario aziendale presenta una *struttura cognitiva* molto vicina a quella dell'ambiente che ha generato una nuova conoscenza, tanto più elevata è la capacità di assorbimento di conoscenze dell'azienda, aspetto che facilita e rende efficace il processo di trasferimento di competenze ed esperienze.

Risulta altresì chiaro che l'importanza della centralità dell'apprendimento comporta l'attribuzione di un ruolo chiave al settore delle risorse umane, aspetto che non può prescindere dall'assegnazione di una particolare attenzione al rafforzamento della dotazione di "capitale umano valorizzato" e allo sviluppo di nuove competenze e conoscenze che possano garantire un costante aggiornamento culturale e professionale interno all'azienda. Solo l'acquisizione di questa consapevolezza può garantire il superamento dei problemi di obsolescenza che i contesti competitivi dinamici inevitabilmente comportano. Gli autori del saggio, hanno anche osservato che le imprese più innovative sembrano essere quelle che risultano essere le migliori nel riconoscere la rilevanza delle informazioni esterne e che riescono ad assimilarle e a elaborarle in maniera corretta.

Ciononostante, le aziende devono affrontare non poche difficoltà nella raccolta delle informazioni, soprattutto a causa della complessità crescente dei contesti tecnologici attuali. Come asserisce Pascal Savioz [2], ricercatore presso l'ETH Center for Enterprise Science (Technology and Innovation Management), in Svizzera, le cause attribuibili a queste difficoltà sono molteplici e alcune di esse sono:

- *Il fenomeno della technology fusion*. Si riferisce al problema dell'aumento dell'interdisciplinarietà dei settori scientifici. La pervasività delle tecnologie informatiche fa sì che la stessa non possa essere assegnata ad un settore scientifico specifico. Di conseguenza, vanno a scomparire le barriere tra i diversi settori scientifici, che si fondono e si intrecciano tra loro rendendo possibile una condivisione di scienze in cui le tecnologie digitali sono pervasive. Ad esempio, lo sviluppo delle biotecnologie rappresenta una perfetta dimostrazione di fusione tra una pluralità di campi scientifici,

come la genetica, la biologia, l'alimentazione, la farmacologia, la chimica e le tecnologie informatiche e l'ingegneria.

- *La portata globale delle tecnologie.* L'integrazione a livello mondiale e il coordinamento delle attività di ricerca e sviluppo, rappresentano un problema sostanziale per la società. La vera sfida consiste nell'ottimizzazione del tempo e del trasferimento delle conoscenze esterne. Le tecnologie di comunicazione possono dare un enorme contributo in tal senso, a condizione che le stesse siano utilizzate seguendo schemi rivolti alla semplificazione del loro utilizzo.
- *La tremenda esplosione di produzione di conoscenze tecniche.* Se consideriamo che, quotidianamente, sono presentate oltre 6.000 pubblicazioni scientifiche, si evince che ogni 10-15 anni la letteratura scientifica si raddoppia. Non bisogna dimenticare che attualmente, oltre 5 milioni di persone lavorano nel settore della ricerca e sviluppo (R&D, Research and Development), cifra che equivale al 90% di tutti gli scienziati vissuti dall'alba dei secoli.
- *Tempi di sviluppo più lunghi che si fronteggiano con cicli di mercato più brevi.* La crescente complessità tecnologica richiede lunghi tempi di sviluppo, e la diversità delle innovazioni provoca cicli di mercato più brevi. Ne deriva che, nel tempo, aumentano le uscite di cassa, mentre il flusso di cassa diminuisce.
- *Aumento dei costi nel settore interno della Ricerca e Sviluppo.* Com'è noto, i costi della R&D aumentano in maniera esponenziale e i tempi di sviluppo di prodotto possono diluirsi negli anni. Se consideriamo che nel solo settore farmaceutico questi costi ammontano a circa 350 milioni di dollari, per un tempo di sviluppo di circa dieci anni, possiamo ben comprendere la problematicità dell'aspetto degli investimenti dedicati allo studio e alle sperimentazioni.

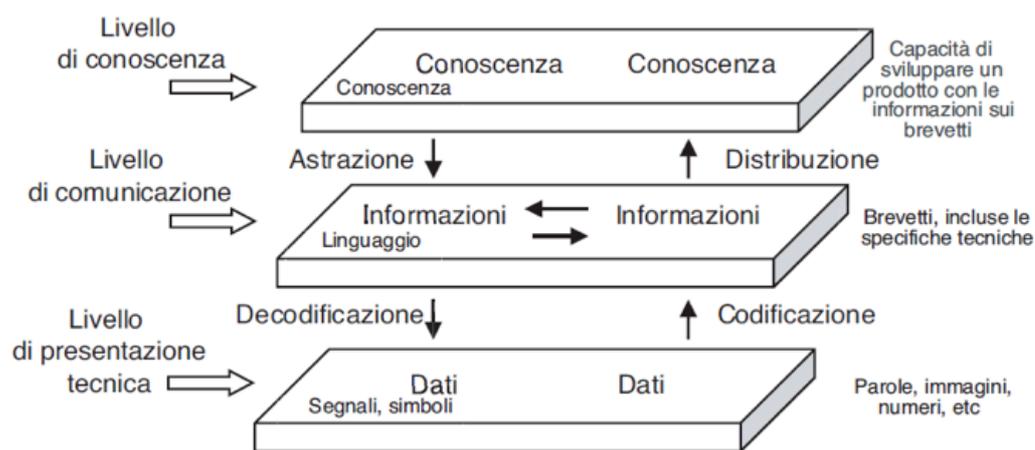
In funzione di quanto evidenziato da Savioz, si può delineare un nuovo percorso in grado di affrontare le sfide del terzo millennio: *la valorizzazione delle informazioni come base per la produzione di conoscenza.* È una vera sfida per le aziende, che si traduce nell'acquisizione della conoscenza per mezzo degli strumenti tecnologici. Abbinando la ricerca scientifica alla letteratura nel settore del management, numerose aziende hanno maturato un corposo interesse verso una nuova dottrina, adottabile per comprendere e utilizzare in maniera ottimale le tecnologie dell'informazione e della comunicazione: la *Technology Intelligence* (TI). Pertanto possiamo definire la TI come "il processo di raccolta e distribuzione di informazioni riguardanti le nuove tecnologie per supportare il processo decisionale all'interno di un'organizzazione" [3].

L'implementazione di un corretto sistema di TI, può consentire, sulla base delle informazioni raccolte, selezionate ed elaborate nel settore delle tecnologie, di fornire un supporto determinate per i processi decisionali. In tal senso, l'organizzazione potrà innanzitutto organizzarsi preventivamente per fronteggiare le possibili minacce che potrebbero manifestarsi in futuro, ma sarebbe anche in grado di attivare delle tecniche rivolte alla valorizzazione degli

strumenti tecnologici utilizzati in azienda. Il crescente interesse verso questa *intelligence discipline* è anche testimoniato dall'aumento del numero dei membri della Strategic and Competitive Intelligence Professionals – SCIP [4], nel corso dell'ultimo decennio.

Per una migliore comprensione del concetto di “...*processo di raccolta e distribuzione di informazioni*”, è opportuno effettuare alcune considerazioni. La TI tende a creare, nell'ambito dell'organizzazione in cui si applica, una *conoscenza* che viene generata dall'analisi dei dati e delle informazioni raccolte. Davis e Botkin [5] le hanno classificate in quattro diverse tipologie: *numeri, parole, immagini e suoni*. Queste tipologie valgono ben poco solo se non sono collegate in un particolare ambiente. Se ciò avviene, si genera una conoscenza specifica. Pertanto, il valore e l'utilità delle informazioni elaborate, da parte di chi le acquisisce, è direttamente riconducibile alla capacità di valutare e utilizzare la conoscenza prodotta dalle stesse. Al termine del processo di assimilazione, analisi e valutazione delle informazioni, si può parlare di conoscenza, che rappresenta “...*l'insieme di esperienze, apprendimento, e competenze degli individui utili per la risoluzione dei problemi*” [6].

Quindi, la conoscenza è costantemente orientata all'azione e alle persone. Da un punto di vista aziendale, secondo Probst [7], “...la conoscenza organizzativa di base corrisponde alla totalità di esperienze individuali e collettive, alla cognizione e alle competenze a cui può accedere l'organizzazione, al fine di risolvere un problema, compresi i dati e le informazioni di riferimento.” Il sistema di trasferimento e interazione tra dati, informazioni e conoscenze, è mostrato in figura 1.



**Figura 1**

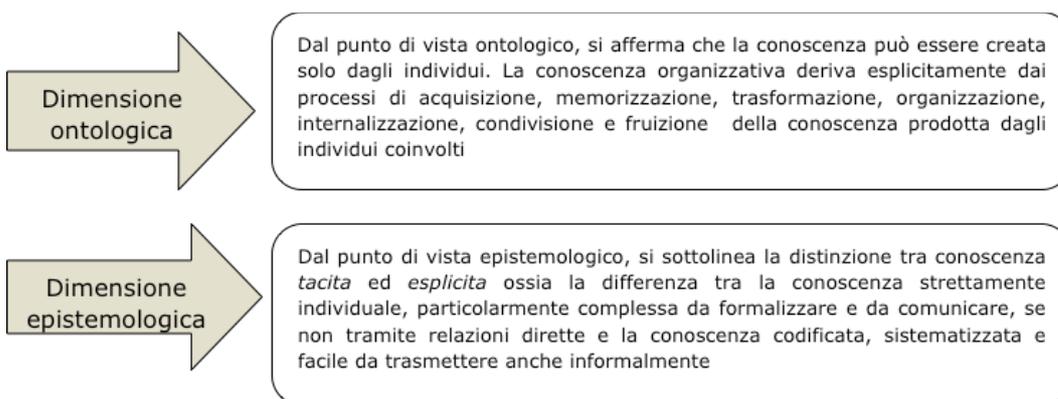
*Interazione e trasferimento di dati, informazioni e conoscenza*

(fonte: Pascal Savioz, “*Technology Intelligence. Concept Design and Implementation in Technology-based SMEs*”, Marzo 2004, Palgrave Mcmillian)

Come possiamo osservare, il trasferimento delle conoscenze avviene in maniera apparentemente lineare tra un individuo all'altro, in funzione della struttura della società in cui vive. In realtà tutte le informazioni trasmesse devono essere *astratte* e *decodificate* per poter essere trasmesse ad un livello inferiore<sup>1</sup>. Pertanto, poiché il ricevitore deve interpretare le informazioni a sua volta ricevute, una parte del significato originale si perde [8].

## 2. Knowledge Management: la gestione della conoscenza

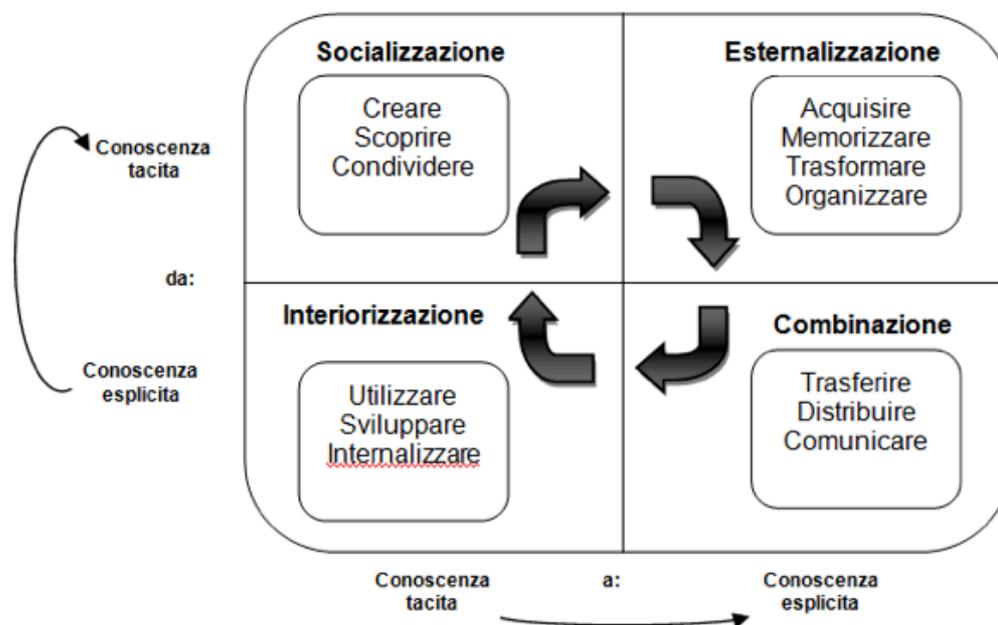
Secondo due studiosi giapponesi [9], Ikujiro Nonaka e Hirotaka Takeuchi, la conoscenza organizzativa si crea attraverso un processo a spirale che prevede l'interazione tra *conoscenza tacita* e *esplicita* a diversi livelli ontologici (da individuale a organizzativo). Tale interazione poggia su quattro modalità di conversione della conoscenza: *da tacita a tacita* (azione sociale), *da tacita a esplicita* (esternalizzazione), *da esplicita a esplicita* (combinazione), e *da esplicita a tacita* (internalizzazione). Per una maggiore comprensione del modello è opportuno innanzitutto chiarire che esso si basa su due *dimensioni*, relative al processo di creazione della conoscenza in azienda:



In funzione di ciò è possibile desumere come possano svilupparsi dei *processi cognitivi* che consentono alla conoscenza tacita di trasformarsi in esplicita e di riprodursi continuamente nel tempo. Quindi nel modello è possibile cogliere l'astrazione dei processi di conversione della conoscenza che si sviluppano quotidianamente e costantemente in un'organizzazione. La conoscenza esplicita viene utilizzata, applicata, internalizzata e sviluppata nel contesto aziendale, consentendo ad ogni individuo di sviluppare la conoscenza tacita. Quest'ultima viene acquisita e riversata, mediante un processo di elaborazione (esplicitazione), su diversi livelli organizzativi, grazie a opportuni strumenti di comunicazione e conoscenza. La distinzione tra conoscenza tacita ed esplicita risulta fondamentale nel processo di creazione della conoscenza, dato che la

<sup>1</sup> L'astrazione, da un punto di vista filosofico, è un metodo logico per acquisire concetti universali ricavandoli dalla conoscenza di oggetti e scenari particolari. La decodifica dell'informazione acquisita viene effettuata dall'individuo in funzione delle sue caratteristiche culturali, ambientali, sociali e mentali.

conoscenza tacita costituisce il fulcro centrale dei processi di innovazione di miglioramento dell'impresa (figura 2).



**Figura 2**

*Il processo di creazione della conoscenza secondo Nonaka e Takeuchi*

Va evidenziato che il termine *conoscenza tacita* o *implicita* è utilizzato nel campo delle discipline che studiano il funzionamento delle organizzazioni, come la sociologia del lavoro e delle organizzazioni, la teoria di impresa, l'economia aziendale e le scienze manageriali, ovvero per identificare le articolate dinamiche relazionali che costituiscono la base per lo sviluppo della conoscenza nelle organizzazioni. Pertanto si tratta di una conoscenza che non è regolamentata o descritta in specifici manuali o guide e non è neppure diffusa mediante strumenti di comunicazione organizzati. È la conoscenza che deriva dalle esperienze personali dell'individuo, dalle *best practices* (pratiche migliori) che ha acquisito, utilizzato e memorizzato nel corso degli anni, insomma dal quel ricco e articolato bagaglio di esperienze professionali e lavorative che possono rappresentare il nettare più pregiato per la valorizzazione delle lavoro svolto in un contesto d'impresa [10]. Il processo di esternalizzazione, consiste nella conversione della conoscenza tacita in *conoscenza esplicita* ed è offerta all'organizzazione per consentire il miglioramento dei processi aziendali e di conseguenza dell'azienda stessa nella sua globalità. Dal modello di Nonaka e Takeuchi il concetto di conoscenza tacita assume un ruolo determinate nella realizzazione del modello di gestione del *knowledge management* aziendale.

Quindi con il termine “gestione della conoscenza”, in senso lato, si è soliti riferirsi a *quel complesso di metodologie e strumenti di comunicazione finalizzati alla gestione e condivisione delle risorse informative acquisite*. Anche se risalente a tempi molto antichi, lo studio delle metodologie rivolte alla gestione della conoscenza, ha raggiunto il periodo di maggior splendore solo con la rivoluzione digitale dei primi anni '80. Non a caso, per la raccolta, lo sviluppo, la memorizzazione e la fruizione della conoscenza, gli individui ricorrono a strumenti e applicazioni che appartengono al mondo dell'Information and Communication Technology (ICT).

L'utilizzo delle strategie di Knowledge Management (KM) è finalizzato al miglioramento delle performance aziendali, del vantaggio competitivo con altri player di mercato e all'ottimizzazione dell'innovazione e della ricerca di settore. La conoscenza assume la connotazione di un *asset strategico*, pertanto il KM può rappresentare la risposta più valida per un'impresa per fronteggiare le sfide del terzo millennio. Non va dimenticato che l'opinione più diffusa vede nel KM lo strumento più efficace per fronteggiare le conseguenze della globalizzazione, soprattutto in funzione dell'esasperazione del fenomeno della concorrenza, che ha condotto un numero sempre maggiore di aziende ad adottare metodologie e pratiche rivolte allo sviluppo del capitale intellettuale della propria organizzazione. Benché l'attenzione delle aziende verso la valorizzazione della conoscenza risalga a tempi non recenti e sicuramente non riconducibili agli studi sul KM, solo negli ultimi decenni la valorizzazione della conoscenza per finalità aziendali ha assunto un ruolo determinante per la scelta delle strategie assunte dai vertici aziendali. All'interno di un'organizzazione, la progettazione, lo sviluppo, la produzione di un bene/servizio articolato e intensamente tecnologico, richiede l'assunzione di una mole di informazioni ragguardevole. Le stesse devono essere correlate da tutti i dati e le notizie che possono consentire di assumere, da parte dell'azienda, un quadro cognitivo completo e esaustivo sulla produzione in corso.

Tutti i processi e le attività aziendali, che interessano il bene/servizio prodotto (progettazione ed organizzazione della produzione e della logistica, gestione della rete commerciale, gestione e assistenza post vendita, analisi dei mercati, specifiche tecniche, customer satisfaction, pianificazione attività e processi, ecc.), a cui si aggiungono le norme, le guide, la raccolta di best practices, congiuntamente alla esperienze individuali delle persone coinvolte nel contesto aziendale, rappresentano la conoscenza aziendale che, se opportunamente custodita e regolata, costituisce il mezzo più rilevante per arricchire e ottimizzare l'efficacia e l'efficienza dei processi d'impresa.

Il modello di creazione della conoscenza di Nonaka e Takeuchi, come abbiamo potuto comprendere, fornisce un contributo rilevante alla nascita della disciplina del Knowledge Management, consentendo a quest'ultimo il passaggio dall'ambito scientifico-accademico a quello gestionale-aziendale. Mediante il passaggio da conoscenza tacita a conoscenza esplicita, si giunge anche alla creazione della teoria della conoscenza organizzativa, meglio identificata con il termine di *spirale di conoscenza*. Il termine trae le sue origini dall'andamento a spirale dei processi di conoscenza (passaggi da tacita ad esplicita e viceversa).

La spirale della conoscenza si basa sull'acquisizione ed utilizzo di tutti i dati "grezzi", che devono subire molteplici processi di raffinazione e valorizzazione per essere trasformati in *saggezza*. L'informazione rappresenta il dato selezionato, organizzato e pronto per essere trasmesso ai settori interessati. La conoscenza rappresenta l'informazione nuovamente elaborata e applicata ai processi e alle attività aziendali. La saggezza è l'essenza della conoscenza, l'esperienza acquisita come valore e ricchezza dell'organizzazione (figura 3).

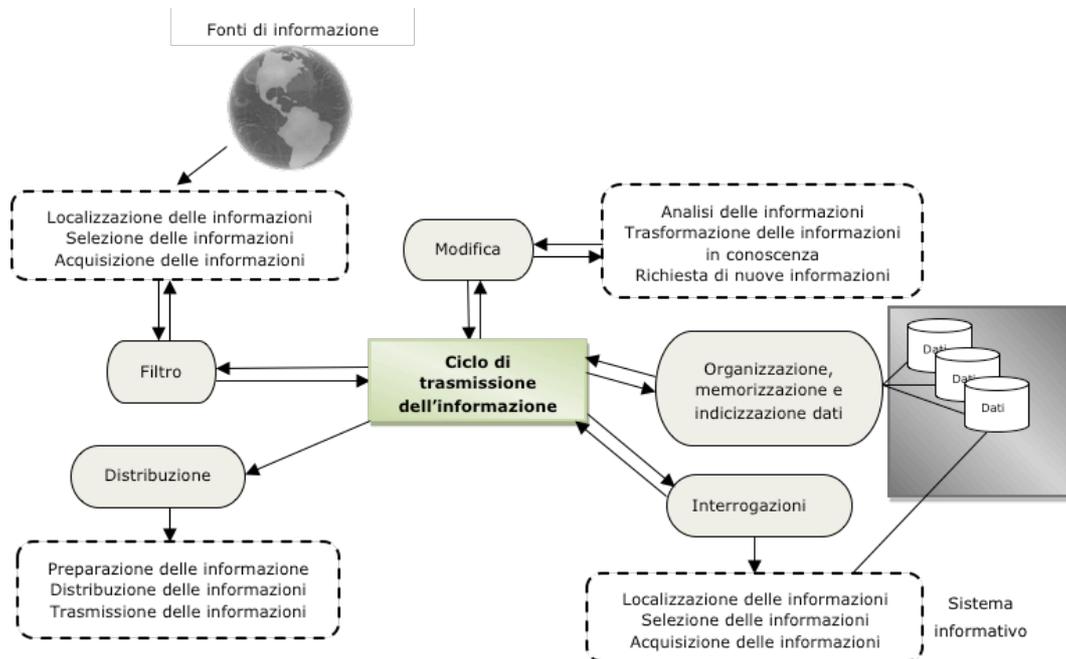


**Figura 3**  
*Spirale della gestione della conoscenza*

È opportuno sottolineare che il KM concentra l'attenzione sulle modalità di fruizione, da parte dell'impresa, della globalità esperienze professionali individuali del personale interno, accumulate nel corso degli anni. È da notare che la possibile resistenza da parte delle risorse umane coinvolte nella condivisione del proprio bagaglio culturale e delle proprie esperienze lavorative, soprattutto in funzione di logiche di gelosia e competizione interna, potrebbe rivelarsi un vero e proprio danno economico per l'azienda. Il danno prodotto sarebbe direttamente riconducibile alla fuga di conoscenza che si avrebbe se un collaboratore decidesse di abbandonare l'azienda, magari accettando l'offerta di un'azienda in diretta competizione con quella di origine. Per questo motivo il knowledge management deve assumere, all'interno dell'aziende, la veste di un *credo filosofico* che vede nella collaborazione e condivisione del lavoro e delle informazioni tra tutti i dipendenti, l'habitat naturale e congeniale per ognuno di essi.

### 3. Il ciclo di trasferimento dell'informazione come base per la produzione di conoscenza

Per consentire la trasformazione delle informazioni grezze in "saggezza", è indispensabile creare un ciclo di trasmissione che preveda una sequenza di passaggi opportunamente strutturati. I cicli di elaborazione/trasmissione delle informazioni si sviluppano in funzione di una serie di circolazioni, cui sono sottoposti i dati, per assumere un valore in grado di arricchire il livello di conoscenza dell'organizzazione. Come si evince dalla figura 4, le informazioni, dopo un'accurata fase di ricerca, localizzazione, assimilazione e selezione, sono convenientemente *filtrate* per poter essere elaborate dall'azienda. Il personale specializzato si occuperà, a questo punto, di analizzare, modificare, organizzare e memorizzare tutte le informazioni raccolte in speciali database memorizzati in sistemi informativi di rete (server), che avranno il compito di conservare e rendere fruibili tutti i dati che saranno oggetto di interrogazioni, modifiche e integrazione da parte dell'utenza autorizzata. Uno dei punti cardine del *transfer cycle informational* è rappresentato dalla capacità di selezionare rapidamente solo le informazioni utili per la produzione di conoscenza. La selezione delle informazioni, prevede la verifica della presenza dei soli dati significativi che possano avere un reale impatto sulle attività aziendali. Pertanto, l'elemento di autentico valore di un modello di TI, consiste nel ricercare e fornire solo le informazioni rilevanti che possono essere di utilità ai vertici decisionali dell'azienda [11].



**Figura 4**  
*Ciclo di trasmissione delle informazioni*

Una particolare attenzione dovrà essere riservata all'identificazione e adozione dei migliori e più idonei dispositivi hardware e software fruibili sul mercato per la realizzazione di un sistema di TI. Solo una corretta ingegnerizzazione dei sistemi informatici può consentire una gestione rapida ed efficace delle informazioni memorizzate. Inoltre il sistema prescelto dovrà garantire la maggiore scalabilità ed interoperabilità con i diversi sistemi informativi esistenti sul mercato e disponibili in Internet, oltre a garantire la più completa interazione con le reti di trasmissione dati. L'importanza dell'approntamento di un *Knowledge Management System* (KMS) è fondamentale principalmente per le fasi di supporto al ciclo dell'informazione e comunicazione all'interno dell'organizzazione. In sostanza, il sistema deve assistere le persone ad esplicitare in maniera completa la propria conoscenza tacita, a reperirla, individuandone i contenuti più importanti e a condividerla con tutti coloro che possono beneficiarne in termini di accrescimento delle proprie competenze. Pertanto, un supporto particolare deve essere dedicato alle seguenti funzioni:

- individuazione e cattura delle competenze collettive;
- individuazione, controllo e monitoraggio della realizzazione degli obiettivi comuni;
- raccolta e integrazione delle conoscenze suddivise.

Attualmente, sul mercato è possibile reperire molteplici tecnologie e applicazioni software in grado di soddisfare le più diverse esigenze riconducibili alla gestione della conoscenza. Tuttavia, essendo il Knowledge Management particolarmente complesso da progettare e strutturare, soprattutto in funzione delle diversità di utilizzo, non si presta alla realizzazione di applicazioni software di tipo standard o non personalizzabili. Pertanto il fatto che non esista una specifica applicazione software per la gestione della conoscenza, ma molteplici tipologie di tecniche, soluzioni, architetture e perfino approcci metodologici, non deve destare particolare stupore e non deve indurre a pensare che non vi sia ancora una maturità nella comprensione dell'importanza del valore delle informazioni. La spiegazione è data semplicemente dalla poliedricità della disciplina, che può comportare l'adozione di molteplici strategie e azioni in funzione delle tipologie di organizzazione a cui si rivolge. Nondimeno è altresì possibile identificare alcuni componenti determinati, da ritenersi sostanzialmente essenziali, che non dovrebbero mai mancare all'interno di un corretto ed efficiente KMS:

- *Enterprise Content Management - ECM* (Gestione dei contenuti dell'impresa). È costituito da un insieme di strumenti che permettono la gestione dei documenti prodotti ed acquisiti dall'organizzazione. Un ECM deve essere munito di *sistemi di form processing* (riconoscimento ottico e intelligente dei caratteri, riconoscimento barcode, riconoscimento scrittura manuale, etc), di *sistemi di gestione del controllo, dell'indicizzazione e della verifica dell'integrità delle informazioni acquisite* (document management, web content management, records management, business process management), di *sistemi di archiviazione e conservazione dei dati* (hard disk, storage unit, NAS), e di *sistemi di distribuzione delle informazioni possedute* (rete aziendale, Internet, Intranet, e-mail, social network, forum).

- *Unified Communication - UC* (Comunicazione unificata). È un sistema che garantisce l'integrazione di servizi di comunicazione in tempo reale, come la messaggistica istantanea (chat), informazioni sulla presenza, telefonia standard e su IP (VoIP), videoconferenza, condivisione di dati, controllo chiamate e riconoscimento vocale, unified messaging (voice mail, e-mail, SMS, MMS, fax). La UC si compone quindi di una serie di prodotti/servizi capaci di fornire delle interfacce in grado di generare informazioni utili per ogni singolo utente. Nella sua accezione, la UC può comprendere anche mezzi di comunicazione come il protocollo di rete TCP/IP che include altre forme di comunicazione quali l'Internet Protocol Television – IPTV<sup>2</sup> e il Digital Signage<sup>3</sup>.
- *Business Process Management System – BPMS* (Sistema di gestione dei processi di business). È un sistema che si occupa della gestione dell'insieme di tutte le attività utili per comporre, migliorare, monitorare e integrare tutti i processi aziendali, per consentire di produrre un processo globale orientato a rendere efficiente ed efficace il business dell'azienda. Le applicazioni software di BPMS permettono di rendere più semplice e rapida la gestione e l'ottimizzazione dei processi aziendali. Per raggiungere questi obiettivi, un BPMS deve essere in grado di monitorizzare la corretta esecuzione dei processi, permettere al personale preposto di condurre analisi e di modificare la tecnologia e l'organizzazione delle basi di dati, produrre dati e informazioni utili per la definizione delle strategie aziendali e per l'identificazione delle decisioni migliori dei vertici aziendali.
- *Skills Certification System, SCS* (Sistema di certificazione delle competenze). È un sistema di mappatura e registrazione delle competenze del personale operante all'interno dell'organizzazione. Consente di identificare le possibili integrazioni e gli sviluppi delle conoscenze personali e collettive in funzione degli obiettivi da conseguire, delle esigenze dell'azienda e delle strategie di business adottate.

Naturalmente, i componenti illustrati non devono essere considerati esaustivi o vincolanti per la realizzazione di moderno KMS. Altri componenti possono essere aggiunti e possono includere anche tecnologie e sistemi di integrazione e interazione tra i diversi elementi che compongono il sistema di gestione della conoscenza. Su quest'ultimo aspetto può basarsi il vero successo di un KMS: l'integrazione tra sistemi e database può giocare un ruolo effettivamente determinante per la valorizzazione delle informazioni acquisite dall'azienda. La comunanza delle ricerche e delle *query* (interrogazioni) sui molteplici database,

<sup>2</sup> *Internet Protocol Television (IPTV)*. È un sistema di tele radiodiffusione dedicato alla trasmissione di contenuti audiovisivi su di un'infrastruttura di rete a pacchetto basato sulla suite di protocolli Internet (TCP/IP).

<sup>3</sup> *Digital Signage*. È una forma di comunicazione di prossimità sul punto vendita o in spazi pubblici aperti o all'interno di edifici, anche nota in Italia come segnaletica digitale, videoposter o cartellonistica digitale, i cui contenuti vengono mostrati ai destinatari attraverso schermi elettronici o videoproiettori appositamente sistemati in luoghi pubblici (fonte Wikipedia).

tra i diversi livelli dell'organizzazione, può consentire non solo di diffondere una conoscenza ad ampio spettro, ma può permettere di comprendere e condividere le scelte e le deduzioni che provengono dall'esame dei dati analizzati. In tal senso si accresce anche l'attendibilità della conoscenza prodotta nell'organizzazione.

#### 4. La valorizzazione delle informazioni: il concetto di Intelligence

La Technology Intelligence si basa sulla ricerca, raccolta ed utilizzo delle informazioni tecnologiche per il miglioramento dei processi decisionali dell'azienda. Pertanto il *base core* della TI è rappresentato dalla capacità di acquisire le informazioni più utili all'azienda per la costruzione e la valorizzazione della propria conoscenza. In tal senso, il problema sostanziale è rappresentato dall'identificazione delle informazioni da considerare *valide e fruibili* sul mercato delle informazioni. In tal senso, risulta consequenziale porsi alcune domande: com'è possibile comprendere quali possono essere i dati interessanti per un'azienda? Come si attua la ricerca, la raccolta, l'analisi e la valorizzazione delle informazioni per la trasformazione in una conoscenza utile all'impresa?

Il passaggio fondamentale che preliminarmente deve anticipare la fase di ricerca delle informazioni, è quello della corretta identificazione dei bisogni informativi dell'azienda. Naturalmente, i settori di interesse devono essere quelli riconducibili all'area in cui opera l'azienda. Di conseguenza, le informazioni da ricercare ed acquisire, devono consentire di fornire elementi nuovi e innovativi in particolare per quanto concerne la pianificazione delle scelte strategiche dell'azienda. Non a caso, la TI è fortemente orientata alle proiezioni sul miglioramento dell'intera organizzazione aziendale.

Pertanto la Technology Intelligence non ha solo il compito di ricercare ed intercettare quelle informazioni che possono consentire di accrescere il valore complessivo dell'impresa, ma deve anche perseguire l'obiettivo di distribuire la conoscenza prodotta ai vertici decisionali, agli individui preposti alla guida delle strutture interne e a tutti quei settori che in cui si rilevi un bisogno informativo.

Le informazioni possono coprire argomenti molto diversi ma è possibile che siano utilizzate per sopperire esigenze diverse e molteplici scopi. Per queste ragioni, all'interno dell'organizzazione, possono essere condotte delle *attività di intelligence* (interne ed esterne) per identificare nuove opportunità e/o possibili minacce, ma soprattutto per preparare le decisioni, che dovranno assumere i vertici decisionali, con i dati migliori, che dovranno consentire di produrre un'immagine dello scenario in cui si trova ad operare l'azienda, più veritiero e attendibile [12].

È opportuno, a questo punto, chiarire meglio il concetto stesso di intelligence. Attualmente sussiste ancora una certa difficoltà di distinzione tra conoscenza e intelligence. Alcuni identificano l'intelligence come sinonimo di conoscenza, altri individuano l'intelligence come un oggetto che interagisce tra l'informazione e la conoscenza e in altre affermazioni di utilizza il termine di "informazioni

analizzate” [13]. Probabilmente, la rappresentazione migliore del significato del termine, ci viene fornita da un intervento del Generale Mario Mori, prefetto ed ex direttore del S.I.S.De.<sup>4</sup>, che asserisce *“La funzione decisionale, quale che ne sia il livello o l’ambito, ha la necessità di disporre di dati informativi idonei per operare le scelte volte a risolvere i problemi di sua competenza. La gestione delle informazioni, quindi, che in una società globalizzata e complessa qual è quella attuale è sempre più elaborata e richiede competenze sempre più diversificate, costituisce una delle fasi più delicate e sensibili del processo che deve portare a definire ogni scelta strategica. Non a caso essa è oggetto di numerosi e accesi dibattiti nei fori nazionali e internazionali. Si moltiplicano, inoltre, organizzazioni specializzate in attività di consulenza, analisi e supporto informativo. Ciò è dovuto al fatto che il bisogno di conoscere va assumendo nel tempo un’inedita centralità, fuoriuscendo dai canonici spazi politico-militari a cui sembrava indissolubilmente legato. L’informazione diventa un bene primario e il suo mercato misura la competitività di attori statuali, sovranazionali e globali. Su queste basi è possibile affermare che l’intelligence è l’insieme delle attività informative volte ad acquisire le conoscenze necessarie a sostenere ogni processo decisionale di natura complessa”* [14].

Come asserisce Robert Steele, tra i più autorevoli esperti al mondo di intelligence, *“...dobbiamo attentamente distinguere tra i dati, che sono costituiti dal testo, il segnale o l’immagine grezza; l’informazione, che è fatta di dati di generico interesse messi insieme; e l’intelligence, che è costituita da informazioni fatte su misura per supportare una specifica decisione da parte di una specifica persona su una specifica questione, in uno specifico momento e luogo”* [15].

Sulla base della definizione di Steele risulta palese come l’intelligence possa estendersi a tutti i quei settori, e tra questi la strategia d’impresa e il mondo dell’economia e della finanza, che interessano il mondo delle aziende. In altri termini, l’intelligence può essere considerata come il *valore aggiunto* che si è capaci di attribuire ad un’informazione, per consentire al decisore di identificare la scelta migliore tra le possibili alternative. **(riquadro 1)**

## **5. L’Intelligence Economica: concetti, interpretazioni, attività**

Cercare di dare una corretta definizione del termine “intelligence economica” non è cosa semplice, soprattutto perché nell’immaginario collettivo l’abbinamento del termine “economico” all’intelligence, scatena rappresentazioni di attività spionistiche di tipo economico, finanziario e industriale. In Italia, all’intelligence economica viene solitamente attribuito un significato che vedrebbe la stessa come legata ad attività *informative* e di *sicurezza*, utili per ricercare alcune particolari tipologie di dati. Interpretazioni diverse si possono riscontrare in altri paesi europei. In Francia, ad esempio, per

<sup>4</sup> S.I.S.De. Il Servizio per le Informazioni e la Sicurezza Democratica (S.I.S.De.) è stato un servizio segreto italiano attivo fino alla riforma normativa del 2007. In seguito ha assunto la denominazione di AISI, Agenzia Informazioni e Sicurezza Interna.

intelligence economica si intende quel complesso di attività riferibili alla ricerca d'informazioni, ma con un'attenzione particolare sui professionisti incaricati del loro reperimento. In altri termini, si tende a focalizzare l'attenzione più sugli "attori" che conducono le azioni di ricerca che non sull'informazione stessa. Gli anglosassoni invece, identificano l'intelligence economica come l'effettiva azione di raccolta e valorizzazione di informazioni di tipo economico. Altro aspetto non irrilevante, è dato dalla scarsa considerazione che viene attribuita, in alcuni paesi, alle attività di intelligence. In Italia, ad esempio, le strutture che si occupano di intelligence sono state spesso associate ad episodi di violenza o scenari equivoci, che hanno contribuito a fornire una bruttissima immagine dell'intelligence all'opinione pubblica. È da ciò che deriva quella sostanziale diffidenza o corposo timore nelle persone quando si pronuncia il termine "intelligence".

Addirittura permane una sostanziale confusione tra "spionaggio economico" e "spionaggio industriale", che alcuni tendono perfino a considerare come equivalenti. Ancora più complessa è la questione che interessa l'aspetto etico delle attività di intelligence economica. Alcuni la considerano al pari di un'attività illegale e quindi non praticabile per nessuna ragione. Nondimeno, di là di ogni considerazione etica o legata a vecchi pregiudizi o schemi mentali ormai datati, l'intelligence economica ha raggiunto, a livello mondiale, un'espansione e un'importanza tale da non poter essere ignorata o di cui potersi privare.

L'immagine negativa che si ha dell'intelligence economica è riconducibile al fatto che le informazioni su aziende, brevetti, tecnologie, mercati e scenari finanziari, sono acquisite con sistemi non sempre legali, grazie al contributo di personaggi o strutture specializzate in attività di spionaggio industriale. A ciò bisogna aggiungere che alcune attività di spionaggio economico possono provocare gravissimi danni ai mercati e all'immagine di aziende pubbliche e private, generando conseguenze rovinose perfino per l'intera economia del paese in cui risiedono.

Evan H. Potter, in una sua pubblicazione [16] di diversi anni fa, citando Samuel Porteous (analista del servizio di intelligence canadese), asserisce che l'intelligence economica racchiude "le informazioni economiche, attinenti la politica o gli scambi, relative a dati tecnologici, finanziari, aziendali e governativi di un Paese, la cui acquisizione da parte di interessi esteri potrebbe, direttamente o indirettamente, influenzare la produttività relativa o la posizione competitiva dell'economia del Paese obiettivo".

Potter inserisce nel contenitore delle informazioni economiche, oltre a quelle commerciali, politiche, finanziarie e istituzionali, anche quelle di tipo "tecnologico", riconducibili alle innovazioni nel settore delle scienze e delle tecnologie. Sono filoni che interagiscono e si intersecano tra loro, dando luogo alla creazione di una base di conoscenza utile non solo per la realizzazione di azioni che mirano all'ottimizzazione della competitività, ma anche alla progettazione di attività di propaganda e, nel caso, di disinformazione. Quindi Potter introduce dei processi di intelligence economica, che operano su differenti livelli di organizzazione (istituzioni, aziende, industria, commercio, università). Il rapporto tra i processi e gli interpreti dell'intelligence economica, si sviluppa su quattro distinti livelli:

1. *primario* (interno all'organizzazione);
2. *intermedio* (reti d'impresa, accordi professionali, convenzioni e associazioni, comunità di servizi, poli di innovazione, etc.);
3. *nazionale* (interazione e accordi con strutture istituzionali e governative, interazioni con governo e centri decisionali politici e corporativi, etc);
4. *transnazionale* (accordi strategici a livello internazionale tra multinazionali e aziende estere, interazioni con governi esteri e strutture istituzionali sovranazionali come l'Unione Europea).

A questo punto, Potter colloca come epicentro del sistema di intelligence economica la *cultura della nazione*, quale fondamento essenziale per consentire al Paese di conseguire miglioramenti sul piano della competitività internazionale. Se questo obiettivo sarà condiviso dall'intera popolazione, maggiore sarà la disponibilità dei protagonisti economici ad accogliere l'intelligence come strumento di innovazione e di crescita del sistema economico del Paese, elemento imprescindibile per l'arricchimento nazionale in termini di occupazione, innovazione multisetoriale e di conseguenza della crescita dello standard di vita. Secondo il docente statunitense, solo una rivoluzione culturale in grado di attribuire all'intelligence economica la giusta importanza e strategicità può condurre verso l'identificazione delle decisioni migliori per superare le sfide del terzo millennio.

## 6. La Technology Intelligence come strumento di valorizzazione delle informazioni

Gli elementi che compongono la Technology Intelligence, sono rispettivamente:

- Technology Management
- Technology Scouting
- Competitive Intelligence
- Strategic Management
- Innovation Management
- Knowledge Management
- Market Intelligence

Il *Technology Management (TM)*, corrisponde ad un insieme di metodologie di gestione delle tecnologie in uso alle organizzazioni, finalizzate al raggiungimento di un vantaggio competitivo. Gli elementi tipici che contraddistinguono il TM, sono la *technology strategy* (che identifica la strategia seguita per l'implementazione delle tecnologie, oppure il ruolo che le stesse devono assumere all'interno dell'organizzazione), la *technology forecasting* (ossia la capacità di individuare le tecnologie maggiormente rilevanti per l'impresa, in particolar modo se si utilizzano tecniche di *technology scouting*), la *technology roadmap* (l'utilizzo di tecnologie in grado di mappare le esigenze del business dell'azienda e delle mutazioni dei mercati di riferimento della stessa), il *technology project portfolio* (il contenitore dei progetti in fase di sviluppo), che

mantiene un filo diretto con il *technology portfolio* (il complesso delle tecnologie in uso). In altri termini, il TM, all'interno di un'azienda, rappresenta lo strumento di comprensione del valore delle tecnologie per il miglioramento dell'organizzazione. L'implementazione delle tecnologie deve produrre necessariamente un impatto migliorativo per l'impresa, altrimenti l'innovazione tecnologica produrrebbe solo un innalzamento dei costi globali. Pertanto lo sviluppo delle tecnologie si giustifica solo se produce "valore", ed è per questo che deve essere considerato dai vertici aziendali come un *centro di valore* e non come un *centro di costo*. L'innovazione tecnologica va introdotta in azienda solo se vi è un effettivo riscontro della sua utilità in una fase o in un periodo in cui si riscontri l'effettiva necessità di investimenti nel settore.

Il *Technology Scouting* è una tecnica che consente di condurre delle ricerche finalizzate alla rilevazione di tutti quei cambiamenti ed eventi che possono produrre conseguenze nell'impresa. È una sorta di metodo di previsione delle evoluzioni delle tecnologie emergenti, dei mezzi di comunicazione e dei piani di ammodernamento tecnologico dell'organizzazione. attivare delle azioni di previsione ed analisi degli scenari, rende l'impresa lungimirante nell'intercettare i cambiamenti in atto, azione fondamentale per garantire la sopravvivenza della società. Le attività di *scouting*, che si basano sull'acquisizione di fonti di informazioni ufficiali e non ufficiali, possono essere condotte dal personale dell'azienda o da consulenti esterni.

La *Competitive Intelligence* è l'attività rivolta alla definizione, raccolta, analisi e distribuzione di tutte le informazioni in grado di produrre una "conoscenza" sui prodotti, servizi, competitor, clienti ed ogni altro elemento utile per fornire un supporto adeguato ai vertici decisionali dell'azienda. Le azioni della CI sono rivolte sostanzialmente alla comprensione e all'analisi di ciò che accade al di fuori dell'azienda, cioè al contesto economico esterno, nel tentativo di comprendere come mantenere sempre alti i livelli di competitività, ridurre i rischi d'impresa, individuare i possibili pericoli che possono ostacolare lo sviluppo, e tentare di anticipare le possibili sfide del futuro. È opportuno sottolineare che si tratta di attività da non confondere con quelle riconducibili all'analisi della concorrenza, che invece focalizzano l'attenzione su di un ventaglio di informazioni molto più limitato.

Lo Strategic Management corrisponde alla gestione delle strategie condotte dall'azienda. In altri termini, è il complesso delle attività rivolte alla definizione degli obiettivi fondamentali dell'organizzazione. La gestione delle strategie da porre in atto nell'organizzazione è affidata solitamente ai vertici della stessa, tuttavia non è insolito che possano partecipare alla definizione delle stesse anche esperti e consulenti aziendali specializzati nella conoscenza e nell'adozione di tecniche e metodologie di definizione degli obiettivi strategici.

L'*Innovation Management* si occupa della gestione dell'innovazione, intesa come azione di modernizzazione dei processi e degli obiettivi, ma può essere riferita anche ad un prodotto/servizio o all'organizzazione dell'azienda. La gestione dell'innovazione deve consentire all'impresa di reagire alle sollecitazioni interne ed esterne rivolte all'implementazione di nuove tecnologie, metodologie, idee, che possano consentire di migliorare l'intera organizzazione.

Alcuni degli strumenti tipici di un sistema di IM, utili per l'individuazione e l'implementazione di proposte e indicazioni migliorative, sono: il brainstorming, la prototipazione virtuale, la gestione del ciclo di vita di prodotto, la gestione delle idee, il project management, la pianificazione delle linee di produzione, la gestione del portfolio, etc.

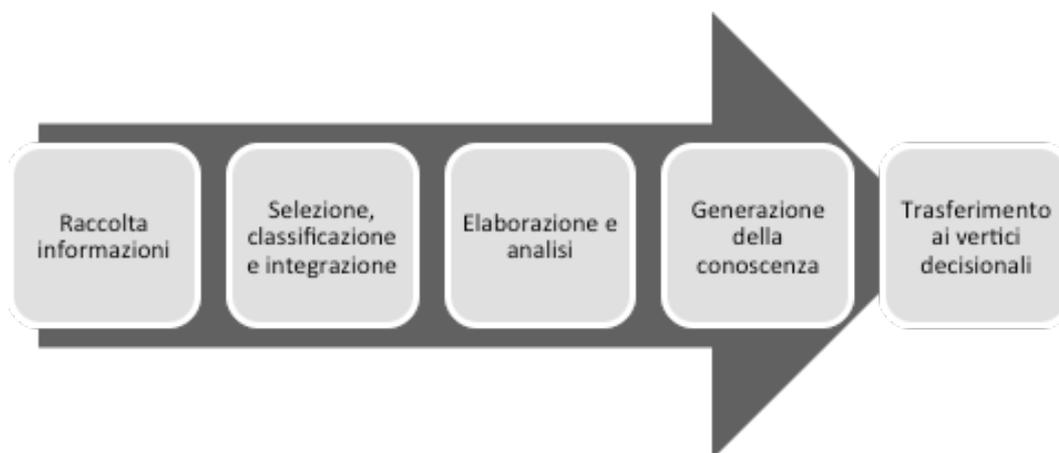
Il *Knowledge Management* può essere inteso come il processo di ricerca, acquisizione, elaborazione e distribuzione della conoscenza. Può essere meglio interpretato come un metodo di approccio multidisciplinare utile per la trasformazione delle informazioni possedute in una saggezza finalizzata all'ottimizzazione del business. Essendo multidisciplinare, può consentire di migliorare settori diversi dell'organizzazione che la utilizza, come quello delle risorse umane, delle tecnologie dell'informazione o delle strategie di business.

Il *Market Intelligence* basa la sua funzione sulla ricerca delle informazioni riconducibili ai mercati in cui opera l'organizzazione. Anche in questo caso, la ricerca delle informazioni è finalizzata alla determinazione delle strategie migliori per la penetrazioni in mercati emergenti, per il miglioramento delle offerte dei prodotti/servizi offerti, per la valutazione di nuove opportunità di mercato, per la valutazione di nuove tipologie di clienti, etc.. E' opportuno evidenziare che il Market Intelligence ha una stretta correlazione con la Business Intelligence che può essere vista come un sistema di supporto per le decisioni da assumere a livello di vertice.

Sulla BI, in funzione dell'interazione con la Technology Intelligence, vale la pena di riportare alcune considerazioni. La prima definizione di Business Intelligence (BI), ci giunge dal ricercatore tedesco impiegato alla IBM, Hans Peter Luhn, che in un articolo del 1958 la definì come "la capacità di cogliere le interrelazioni dei fatti presentati in modo tale da orientare l'azione verso un obiettivo desiderato". Nel 1989, Howard Dresner, analista di Gartner Group, ha rivisitato il termine dandogli un'accezione più tecnologica, assumendo che "La Business Intelligence descrive la capacità dell'impresa di accedere ed analizzare le informazioni, spesso contenute in un Data Warehouse, e di analizzarle per sviluppare conoscenze e comprensione che conducono, con consapevolezza, al miglioramento dei processi decisionali. BI include i seguenti strumenti: query ad hoc, realizzazione di reportistica, sistemi di supporto alle decisioni (Decision Support Systems, DDSs), sistemi informativi direzionali (Executive Information Systems, EISs) e, spesso, le tecniche come l'analisi statistica e l'elaborazione analitica in linea (On Line Analytical Processing, OLAP). Una definizione più semplicistica, ma nel contempo più adeguata, potrebbe definirla come l'insieme di processi, metodologie e strumenti tecnologici in grado di produrre un valido ed efficace strumento di supporto alle decisioni, mediante la trasformazione dei dati aziendali in informazioni utili (figura 5).

Di là dalle definizioni riportate, la BI, nel corso degli anni, ha assunto significati diversi, riferibile alla gestione dei processi di ricerca/analisi delle informazioni, alla gestione dei sistemi informatici e per indicare la globalità delle informazioni, gestite dal management, per la conduzione delle attività aziendali. Alcuni l'hanno definita come il complesso delle applicazioni utili per il supporto alle decisioni aziendali, ma la BI è qualcosa di più complesso, una sistema in grado

di creare innovazione e miglioramento della qualità in diversi contesti di impresa. È un insieme di modelli, procedimenti, processi, professionalità e tecnologie che consentono di rendere l'azienda costantemente competitiva.



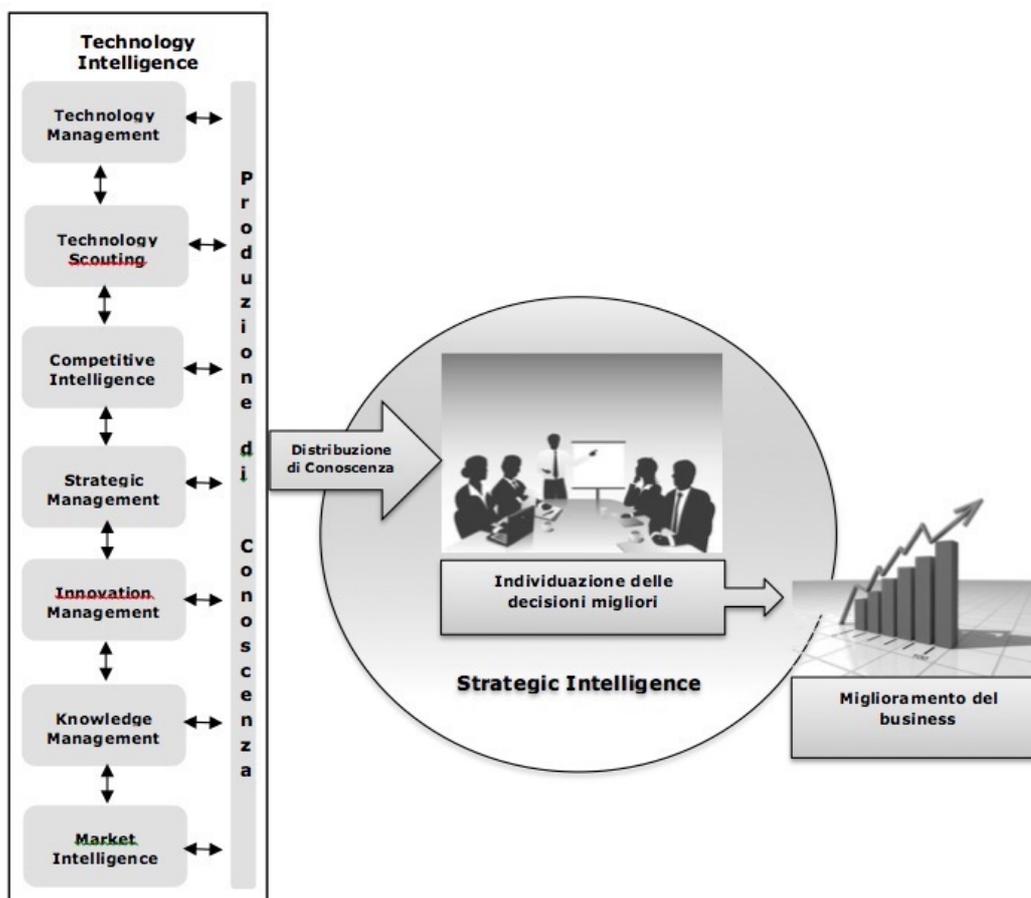
**Figura 5**  
*Attività di Business Intelligence*

Generalmente tutte le fasi che compongono un sistema di BI, sono attualmente gestite attraverso l'utilizzo di piattaforme hardware/software, opportunamente realizzate in funzione delle specifiche esigenze delle aziende in cui sono implementate. Tuttavia è opportuno rilevare che la semplice adozione di un sistema di BI, non è sufficiente a garantire l'efficacia in termini di competitività. Esso può garantire ai vertici aziendali la fruizione di un sistema di conoscenza in grado di soddisfare qualsiasi bisogno informativo, ma senza la presenza di professionalità adeguatamente formate per interpretare in maniera intelligente le informazioni prodotte, qualsiasi sistema di BI perderebbe ogni efficacia.

Per questo motivo l'efficienza di un sistema di Business Intelligence dipende soprattutto dalla genialità, lungimiranza, cultura e intuitività del personale preposto alla sua gestione. Ad esempio, uno degli errori più comuni riscontrabili nell'adozione di un sistema di BI è dato dalla propensione del management aziendale ad accettare le imposizioni e i limiti che i sistemi di BI esistenti sul mercato, spesso impongono. Devono essere le esigenze del business, gli obiettivi prefissati, i requisiti del management e le occorrenze degli analisti, a definire le specifiche per la progettazione e realizzazione di un sistema di BI che si adatti perfettamente al contesto aziendale in dovrà operare. Di conseguenza, la tecnologia deve assumere una posizione subordinata e di adattamento alle esigenze del management, e non il contrario.

Come abbiamo potuto comprendere, la TI se implementata in maniera corretta, è in grado di assumere il ruolo di *produttore di conoscenza*, una funzione essenziale per il management aziendale sempre più esposto a rischi derivanti da dati errati e condizionamenti di carattere informativo. In funzione del pericolo di una disinformazione crescente, l'attività di ricerca ed elaborazione delle

informazioni più preziose per i vertici aziendali, può avvalersi di una vera e propria metodologia di intelligence, nota con il termine di Strategic Intelligence.



**Figura 6**  
*Il Flow-chart dei processi della Technology Intelligence*

La *Strategic Intelligence* (STRATINT) focalizza la sua attenzione sulla ricerca, raccolta, elaborazione, analisi e diffusione di *conoscenza strategica*, indispensabile per la pianificazione di strategie economiche, politiche e militari, sia a livello nazionale che internazionale. In un contesto di impresa o di organizzazione complessa, la differenza sostanziale che sussiste tra l'intelligence strategica e quella tattico-operativa, risiede nella capacità della prima di comprendere e commentare l'impatto dei cambiamenti di tendenza, l'alterazione degli ambienti e degli scenari, le modificazioni delle attività nel corso del tempo, con il preciso scopo di alimentare una consapevolezza aggiornata ed costantemente utile ai processi di definizione delle strategie aziendali (figura 6).



Don McDowell, nel suo trattato [19] sulla STRATINT, definisce la pratica dell'intelligence come una "forma d'arte", che si evolve continuamente con particolare dinamismo (figura 7). L'intelligence strategica è oggetto di continue messe a punto, che si accompagnano all'evoluzione delle tecnologie e delle metodologie di gestione delle informazioni. Identifica, inoltre, ben sette fasi distinte per la creazione di un modello di intelligence strategica:

1. *Strategic Intelligence Planning* (Pianificazione delle strategie di intelligence). Un requisito essenziale per realizzazione di progetto d'intelligence strategica è quello di acquisire una consapevolezza dell'ambiente in cui devono insistere le attività di intelligence. Ciò si traduce in un lavoro di studio, da parte delle risorse umane dedicate, sulla globalità delle problematiche esaminate e non sui dettagli o i particolari degli scenari, situazioni o singole informazioni esaminate. Se l'obiettivo è quello di individuare cambiamenti e/o minacce, allora l'approccio deve essere di tipo omnicomprensivo, rivolto allo sviluppo di un modello concettuale in grado di esaminare il quadro d'insieme.
2. *Defining the problem correctly* (Definizione corretta dei problemi). La fase di pianificazione di un progetto di intelligence richiede, come base di partenza, che il personale preposto alle attività comprenda in maniera completa e chiara il compito che li attende. Le problematiche riconducibili alle questioni strategiche, spesso possono produrre equivoci o imprecisioni sulle reali esigenze di acquisizione di conoscenza da parte dei vertici decisionali. Per questo motivo è essenziale che siano correttamente recepite le reali esigenze dei *consumatori* di intelligence, per evitare sprechi in termini di risorse umane e finanziarie.
3. *Developing a strategic collection plan* (Sviluppo di un piano strategico di raccolta). La raccolta dei dati rappresenta una delle fasi più complesse del ciclo di intelligence, in funzione dell'affidabilità, precisione, pertinenza, tempestività che l'azione deve garantire. Inoltre essa non deve essere considerata come un'attività di tipo "start-and-stop", ma deve tendere ad essere un *continuum*, un'azione da condurre costantemente nel tempo e che tende a modificazioni istantanee in funzione delle più piccole e mutevoli percezioni di informazioni provenienti da fonti diverse. L'attività di raccolta delle informazioni deve essere ampia e omnicomprensiva, ma non deve degenerare nell'accumulo frenetico di tutti i dati disponibili. Ciò condurrebbe inevitabilmente ad una dispersione di tempo e risorse umane che non avrebbe alcun senso per le attività di intelligence. La raccolta deve basarsi sull'applicazione di un modello concettuale che preveda un'attività di analisi preliminare delle tipologia delle informazioni veramente utili da acquisire.
4. *Generating working hypotheses* (Generazione di ipotesi di lavoro). È la fase in cui sono generate ipotesi e punti di vista su scenari, persone, situazioni e settori di interesse. Prevala la natura umana, cioè la capacità degli analisti di sviluppare idee e visioni in grado di superare il semplice contenuto dell'informazione assorbita. Le idee, le opinioni e i pareri (*working hypothesis*) sono espressi come probabili risposte, osservazioni

o giudizi, che potrebbero utilmente descrivere una particolare situazione, spiegare le cause dei cambiamenti di tendenza o influire sulle previsioni di future evoluzioni.

5. *Acquiring additional specific data* (Acquisizione di dati specifici aggiuntivi). In funzione dell'ampiezza della fase di acquisizione delle informazioni, spesso si rende necessario sviluppare ulteriori azioni di raccolta delle stesse. Ciò è dovuto al fatto che le attività di analisi possono concentrarsi su specifiche aree di interesse, su tipologie particolari di dati e di fonti di provenienza. Lo scopo di queste estensioni di raccolta di informazioni, è di acquisire ulteriori dati caratteristici che possano consentire di testare e validare la formulazione di ipotesi, teorie e punti di vista.
6. *Processing-recording, collating and evaluating* (Elaborazione-registrazione, raccolta e valutazione). L'intelligence strategica, per sua natura, incorpora metodi e tecniche diverse per quanto concerne la registrazione, il raffronto e la valutazione delle informazioni. Il ciclo di intelligence, in funzione delle organizzazioni che ne fanno uso, richiede approcci diversi e flessibili per svolgere tutte le sue funzioni. Ogni *collection plan*, è unico nel suo genere, in termini di dimensioni, di tipologia di dati, di varietà e caratteristiche delle fonti e di strategia di analisi. Quest'ultima risulta fondamentale per determinare le procedure più idonee per la registrazione, la collazione e la valutazione dei dati. Proprio in quest'ultima fase è possibile notare la differenza maggiore tra l'intelligence tradizionale e quella strategica. Nel classico processo di intelligence, la valutazione dei dati si basa sulla verifica dell'affidabilità e credibilità delle informazioni. Nell'intelligence strategica si tende ad effettuare una valutazione preliminare, frutto della capacità "mentali" dell'analista, che si esprime nelle sue doti di analisi analitica e interpretativa dei dati acquisiti.
7. *Integration, analysis and interpretation* (Integrazione, analisi e interpretazione). La fase classica di analisi si basa sulla selezione, integrazione e interpretazione dei dati elaborati. L'analisi strategica si fonda su due elementi distinti, che costituiscono gli elementi di sviluppo della conoscenza: l'ambiente e il risultato. Il primo è riconducibile al contesto in cui si svolge l'attività di intelligence. La raccolta e il trattamento delle informazioni è un processo quantificabile e controllabile. Lo è di meno l'ambiente in cui si svolge il lavoro di analisi. È in questo "habitat" che si gioca il futuro della valorizzazione delle informazioni. Il lavoro degli analisti deve essere valutato nella sua globalità, ma con una particolare attenzione alle singole attività condotte dai singoli individui e dagli eventi e i risvolti che caratterizzano le attività di gruppo. Il risultato, descrittivo o esplicativo che sia, deve produrre una conoscenza in grado di consentire la formulazione di eventi futuri. I due elementi, sinergicamente, devono costituire il nucleo di un *sistema di intelligence* produttivo ed efficace, di ricerca avanzata e di analisi strategica finalizzato alla raffinazione delle ipotesi e lo sviluppo di previsioni.

## 7. Un modello di analisi multiforme delle informazioni per la TI: il modello PESTLE

Come abbiamo potuto comprendere, le ragioni che spingono un'organizzazione ad utilizzare un sistema di intelligence, sono direttamente connesse all'esigenza di valorizzare le informazioni per fornire nuovi e aggiornati elementi di conoscenza a coloro che hanno il compito di assumere decisioni strategiche e di rilevante impatto sul futuro dell'azienda. Per garantire la fruizione di informazioni attendibili e valorizzabili, è opportuno identificare chi sono i soggetti o le strutture che necessitano di tali informazioni, quali siano quelle di cui hanno effettivamente bisogno e come le stesse debbano essere comunicate.

È opportuno chiarire che le attività di intelligence tecnologica, contrariamente a quanto si possa immaginare, non debbono focalizzarsi sulla ricerca di informazioni "tecniche", ma devono essere orientate a tutti i settori e i contesti che possono interessare lo scenario in cui l'azienda opera.

Una delle metodologie più diffuse per la ricerca e l'analisi delle informazioni è il modello di analisi PESTLE (Political, Economic, Social, Technological, Legal, Environmental). È un sistema in grado di esaminare le principali forze macro-ambientali che interagiscono nell'ambito in cui opera l'impresa e che permettono di definire una cornice cognitiva da cui estrapolare le variabili che possono contribuire a fornire informazioni utili per i processi decisionali e per l'identificazione delle scelte strategiche aziendali. Come l'acronimo evidenzia chiaramente, il modello si basa sulla valutazione di alcune variabili macro-ambientali:

- *La politica.* In che modo la politica di un governo può influenzare l'economia o un particolare settore industriale o commerciale? La politica di un dato paese è sempre più incentrata sull'aumento del livello di tassazione delle imprese? Queste sono solo alcune delle domande che possono interessare i vertici di un'azienda. Il fattore politico influenza le politiche fiscali, il commercio, i sistemi di tariffazione, l'andamento dei mercati finanziari, le attività di lobbying, i rapporti commerciali internazionali, etc. Per questo motivo le informazioni di carattere politico possono costituire un elemento di conoscenza fondamentale per un'organizzazione, soprattutto se essa opera su mercati esteri.
- *L'economia.* Il fattore economico risulta determinante per l'impatto diretto che produce sull'infrastruttura industriale e commerciale di un paese. Ad esempio, un aumento del tasso di inflazione, che produce modificazioni sul potere d'acquisto della moneta, genera conseguenze dirette sulla domanda/offerta di beni e servizi prodotti dalle imprese. Le modificazioni sui tassi di interesse, i tassi di cambio sui livelli di crescita economica, il reddito pro-capite, sono tutti elementi in grado di influenzare enormemente le aziende e le attività imprenditoriali. L'acquisizione di informazioni aggiornate su questi aspetti, costituisce un'azione fondamentale per l'adozione di misure preventive o proattive.

- *Il sociale.* Le informazioni sugli aspetti sociali, come le tendenze culturali, la storia della nazione, la demografia, le statistiche sulla popolazione, i fattori etnici e religiosi, la salute dei cittadini, la moda, le abitudini e gli stili di vita, le classi socioeconomiche, gli influssi dei media sui consumi, le opinioni dei consumatori, i livelli occupazionali, sono solo alcune delle preziosissime indicazioni che possono accrescere il repository informativo di un'organizzazione. L'aspetto sociale è determinante per le politiche aziendali di qualsiasi impresa, indipendentemente da settore in cui opera.
- *Le tecnologie.* In quella che viene definita come cyber-società, cioè quella collettività che attualmente è fortemente influenzata da strumenti informatici e digitali, le tecnologie ICT recitano un ruolo sostanziale. La conoscenza delle più avanzate e recenti innovazioni nel settore, la produzione di brevetti, l'abbattimento del digital-divide, sono delle informazioni essenziali per lo sviluppo e l'adozione delle tecnologie informatiche nei processi aziendali. La consapevolezza tecnologica è sinonimo di avanguardia e lungimiranza per la proiezione dell'azienda in un futuro sempre più dominato da mercati globalizzati e scenari competitivi.
- *L'aspetto legale.* L'aspetto legale sta assumendo un'importanza sempre più crescente nelle organizzazioni. Ciò è dovuto soprattutto all'inarrestabile fenomeno dell'internazionalizzazione dei mercati e dalla necessità di regolamentare, a livello mondiale, le norme che caratterizzano gli aspetti commerciali, economico-finanziari e sociali dei vari paesi. La conoscenza delle normative e della legislazione corrente, la tutela del diritto al lavoro, la tutela dei consumatori, gli organismi di regolamentazione legislativa, le normative ambientali e di sicurezza del lavoro, risultano essenziale per evitare di incorrere in problematiche che potrebbero produrre conseguenze negative, se non disastrose, per qualsiasi tipologia di azienda.
- *L'ambiente.* Soprattutto negli ultimi anni, l'attenzione del mondo economico sull'ambiente, ha subito un rapido incremento. La Green Economy<sup>5</sup>, si è imposta come un nuovo modello di sviluppo economico per il terzo millennio, ma anche le recenti mutazioni climatiche e i fenomeni idrogeologici che stanno interessando l'intero pianeta, stanno influenzando gli studiosi di economia politica per quanto concerne la creazione di nuovi modelli di sviluppo economico. È per questo che le analisi ambientali sono determinanti per i contesti di sviluppo aziendale.

Gli elementi esaminati, sono da considerarsi fondamentali per i business di qualsivoglia azienda o struttura operante in un qualsiasi settore commerciale e industriale, indipendentemente dalla sua ubicazione geografica [20]. Il modello

<sup>5</sup> *Green Economy.* È un modello teorico di sviluppo economico che prende origine da una analisi econometrica del sistema economico dove oltre ai benefici (aumento del Prodotto Interno Lordo) di un certo regime di produzione, si prende in considerazione anche l'impatto ambientale cioè i potenziali danni ambientali prodotti dall'intero ciclo di trasformazione delle materie prime a partire dalla loro estrazione, passando per il loro trasporto e trasformazione in energia e prodotti finiti fino ai possibili danni ambientali che produce la loro definitiva eliminazione o smaltimento (fonte Wikipedia).

di analisi PESTLE può essere considerato come il nucleo fondante su cui costruire gli obiettivi e le strategie di un'organizzazione. Naturalmente le informazioni riconducibili ai diversi elementi possono avere un peso o un valore diverso, soprattutto in funzione della specificità dell'impresa, ma nella loro globalità contribuiscono tutte a formare quel contenitore informativo essenziale per generare quella conoscenza di cui qualsiasi organizzazione necessita per la sua sopravvivenza.

## 8. Conclusioni

Informazioni e intelligence. Sono due termini che identificano l'elemento e la metodologia su cui si baseranno le attività di tutte le organizzazioni del mondo nel prossimo futuro. In un mondo sempre più condizionato dalle tecnologie digitali e da applicazioni fruibili in Rete, che oggi consentono di eliminare quella linea di confine che finora aveva separato il mondo reale da quello virtuale, i dati costituiscono l'unico combustibile in grado di alimentare il motore economico, finanziario, industriale, politico e militare di un paese. Chi non sarà in grado di gestire le risorse informative, non sarà in grado di superare le sfide del terzo millennio.

Sui tempi di acquisizione e valorizzazione delle informazioni si giocheranno i destini delle nazioni, delle aziende, della politica e del nostro stesso tessuto sociale. Alcuni lungimiranti politologi ed economisti, già definiscono le informazioni come "l'oro nero" del terzo millennio. Per certo chi si rivelerà pronto alla loro gestione, potrà contare su una "supremazia cognitiva" che si rivelerà strategicamente determinate in qualsivoglia settore o contesto geopolitico. L'aspetto fondamentale è quello di non giungere impreparati a questo scenario, dato che l'incomprensione di questo scenario futuro potrebbe rivelarsi fatale.

### Riquadro 1: Attività di Intelligence

Nell'intelligence vanno ricomprese tutte le attività informative che possono consentire di supportare i processi decisionali. Ad essa quindi viene affidato il compito di confortare e guidare il management aziendale per l'assunzione delle decisioni più giuste, idonee al conseguimento degli obiettivi prefissati. Per quanto concerne le fasi che solitamente caratterizzano il processo di gestione delle informazioni, il modello di intelligence annovera:

- *Pianificazione e indirizzo.* I vertici decisionali e direttivi individuano gli obiettivi strategico-informativi ritenuti essenziali e sostanziali per il supporto alle decisioni. Gli obiettivi sono definiti attraverso la definizione degli *Elementi Essenziali di Informazione (EEI)* e *Altri Elementi di Informazione (AEI)*. Il primo gruppo è riconducibile alle situazioni "principali" o "fondamentali" su cui concentrare la raccolta di informazioni: il secondo è dato da tutte le informazioni collaterali ma sostanziali agli elementi essenziali.
- *Ricerca e raccolta.* È la fase di ricerca delle informazioni utili per gli obiettivi prefissati, cui segue quella di raccolta e catalogazione delle informazioni acquisite. Le notizie possono essere reperite mediante l'utilizzo di fonti diverse, ma identiche a quelle utilizzate dai servizi di intelligence istituzionali: Intelligence umana (HUMINT), Intelligence delle immagini (IMINT),

Intelligence dei segnali (SIGINT), Intelligence delle misurazioni e delle caratteristiche (MASINT), Intelligence tecnica (TECHINT), Intelligence delle fonti aperte (OSINT).

- *Selezione ed elaborazione.* In questa fase le informazioni raccolte sono selezionate in base alla loro effettiva utilità per il processo di produzione della conoscenza. Sono quindi classificate, valutate e correlate tra loro in funzione dell'effettiva utilità che possono dare al processo di crescita cognitiva dell'organizzazione.
- *Analisi e generazione di conoscenza.* Le informazioni "strutturate" vengono quindi sottoposte ad un processo di elaborazione che le porta ad assumere un valore in termini di affidabilità e qualità, elementi indispensabili per la produzione di una *conoscenza adeguata*.
- *Distribuzione.* È l'ultima fase del processo di intelligence. La conoscenza prodotta deve essere distribuita alle strutture preposte per poter concorrere attivamente al miglioramento delle strategie messe in atto dall'organizzazione. La distribuzione della conoscenza va effettuata con grande attenzione, in funzione delle competenze, delle aree di interesse e dei ruoli assunti dalle diverse strutture.

Da un punto di vista economico, la gestione delle informazioni (*intelligence economica*) è vista come quel complesso di condotte messe in atto per acquisire, valorizzare, divulgare e scambiare delle informazioni per trasformarle in un'azione che mira al miglioramento della competitività aziendale.

Tutte le attività di raccolta delle informazioni aziendali, mirano all'adattamento a nuovi scenari commerciali, all'anticipazione delle azioni di aziende concorrenti, all'acquisizione di nuovi mercati e alla modificazione delle condizioni commerciali e industriali in funzione delle metamorfosi dei mercati globalizzati. Sono assimilabili ad azioni di intelligence economica anche quelle conducono all'adesione a *lobby* o l'assunzione di attività come il *benchmarking*. Possiamo quindi definire *l'intelligence economica come un accomodamento dell'intelligence alle problematiche delle attività economiche, con una particolare attenzione alle questioni della concorrenza e dei mercati globalizzati*.

L'attenzione riservata in questi ultimi anni all'intelligence economica deriva in buona parte dalla consapevolezza, da parte dei protagonisti economici, dell'importanza assunta dall'informazione a livello mondiale e dal fatto che la si possa considerare come il "bene" più prezioso da utilizzare per competere nei nuovi scenari economici iperconcorrenziali. Nel mondo economico, si sta quindi assistendo ad una profonda metamorfosi culturale soprattutto per quanto concerne la formazione, la comunicazione, l'innovazione e le metodologie e le tecniche da utilizzare per il miglioramento dell'impresa. In tal senso, una nuova figura, da molti ritenuta ormai indispensabile per garantire lo sviluppo e la stessa sopravvivenza dell'impresa, è il professionista dell'informazione o Data Scientist (scienziato dei dati), figura professionale di grande rilevanza che si rivelerà fondamentale, nel prossimo futuro, per qualsiasi contesto aziendale.

## Bibliografia

- [1] W. M. Cohen, D. A. Levinthal (1990), "Absorptive Capacity: A New Perspective on Learning and Innovation", *Administrative Science Quarterly*, Vol. 35, No. 1, Special Issue: Technology, Organizations, and Innovation.
- [2] P. Savioz (2004), "Technology Intelligence. Concept Design and Implementation in Technology-based SMEs", Palgrave Mcmillian.
- [3] L. Mortara, C. Kerr, D. Probert, R. Phaal (2009), "Technology Intelligence. Un modello per monitorare gli sviluppi scientifici e tecnologici per identificare minacce e opportunità", (traduzione e adattamento di Milena Mortata e Silvia Fossati), Strategie e Innovazione.
- [4] <http://www.scip.org/> (ultimo accesso novembre 2014)
- [5] J. Botkin, S. Davis (1994), The Coming Knowledge-Bond Business, Harvard Business Review.
- [6] P. Savioz (2004), "Technology Intelligence. Concept Design and Implementation in Technology-based SMEs", (pag. 7), Palgrave Mcmillian.
- [7] G. Probst, S. Raub, K. Romhardt (2003), "Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen", FAZ - Gabler Edition, Gebundene Ausgabe.
- [8] A. Teti (2013), *Technology Intelligence. Ottimizzazione delle tecnologie aziendali*, Pearson Editore.
- [9] I. Nonaka, H. Takeuchi (1995), "The knowledge-creating company", Oxford University Press, (trad.it. 1997, Guerini e Associati).
- [10] A. Teti (2007), *Management dei servizi IT*. pag. 1-235, Il Sole 24 Ore.
- [11] A. Teti (2012), *Il potere delle informazioni. Comunicazione globale, Cyberspazio, Intelligence della conoscenza*. pag. 1-286, Il Sole 24 Ore.
- [12] A. Teti (2012), *Il potere delle informazioni. Comunicazione globale, Cyberspazio, Intelligence della conoscenza*. pag. 1-286, Il Sole 24 Ore.
- [13] L. M. Fuld (1994), "The New Competitor Intelligence. The Complete Resource for Finding, Analyzing, and Using Information about Your Competitors", John Wiley & Sons.
- [14] <http://gnosis.aisi.gov.it/Gnosis/Rivista7.nsf/servnavig/7> (ultimo accesso novembre 2014)
- [15] R. D. Steele (2002), "Intelligence. Spie e segreti in un mondo aperto" Rubbettino, Soveria Mannelli.
- [16] E. H. Potter (1998), "Economic Intelligence and National Security", Carleton University Press.
- [17] M. Maccoby (2001), *Successful Leaders Employ Strategic Intelligence*, Research Technology Management, Volume 44. No. 3. May-June, pp. 58-60. *The Productive Narcissist*, Broadway Books, 2003, chapter 4.
- [18] M. Maccoby (2007). *The Leaders We Need, And What Makes Us Follow*, Harvard Business School Press, 2007, chapter 5.

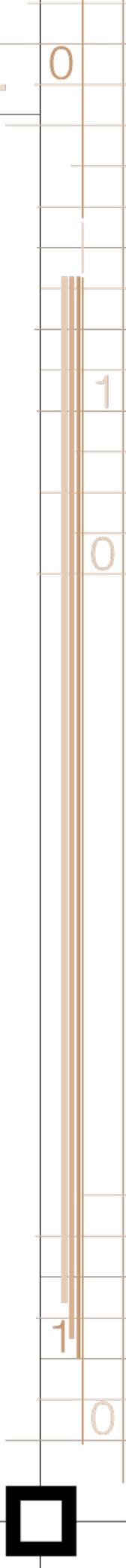
[19] D. McDowell (1997), *Strategic Intelligence & Analysis, Guidelines on Methodology & Application*, The Intelligence Study Centre.

[20] A. Teti (2009), "Il futuro dell'Information & Communication Technology Technologie, timori e scenari della "global network revolution", pag. 1-162, Springer-Verlag.

## Biografia

**Antonio Teti** è docente di Tecnologie di persuasione nel Cyberspazio al Corso di Laurea Magistrale in Psicologia Clinica e della Salute presso l'Università "G. D'Annunzio" di Chieti-Pescara. Svolge attività didattiche e seminariali nei settori IT Security e IT Governance al Corso di Laurea Magistrale di Economia Aziendale nella stessa università. Esperto di Cyber Intelligence, ICT Management e ICT Security, ha accumulato un'esperienza trentennale in studi, ricerche e consulenze in aziende pubbliche, private e istituzioni governative. È stato docente di Computer Science presso diverse università italiane tra. Autore di numerose pubblicazioni scientifiche collabora con diversi quotidiani, riviste e periodici del settore. Tra le sue pubblicazioni: EUCIP, Il manuale per l'informatico professionista (Hoepli); Business and Information System Analyst, Il Manuale per il Manager IT (Hoepli); Network Manager, Il manuale per l'Amministratore di Reti e Sistemi (Hoepli); Management dei servizi IT: dal modello ITIL all'ISO/IEC 20000 (Il Sole 24 Ore); Sistemi informativi per la Sanità, ECDL Health (Apogeo); Il futuro dell'Information & Communication Technology: tecnologie, timori e scenari della global network revolution (Springer Verlag); PsychoTech, il punto di non ritorno. La tecnologia che controlla la mente (Springer Verlag); Il potere delle Informazioni. Comunicazione globale, Cyberspazio, Intelligence della conoscenza (Il Sole 24 Ore); Technology Intelligence. Ottimizzazione delle tecnologie aziendali (Pearson).

email: teti@unich.it



# TORCIA:

## una piattaforma collaborativa per la gestione delle emergenze

Chiara Francalanci - Paolo Giacomazzi

### Sommario

*TORCIA è un progetto finanziato da Regione Lombardia per lo sviluppo di una piattaforma che valorizzi l'utilizzo dei social media nei processi di gestione delle emergenze. Tramite la raccolta continua e in tempo reale di informazioni dai principali social media, TORCIA è in grado di individuare e localizzare geograficamente situazioni di potenziale emergenza, supportare gli operatori istituzionali con un cruscotto decisionale e favorire la realizzazione di un processo collaborativo che integra i contributi di cittadini e istituzioni tramite una mobile app anch'essa integrata con la piattaforma.*

### Abstract

*TORCIA is a project funded by Regione Lombardia. The main goal of the Torcia project is the development of a platform that leverages social media in emergency management processes. With a continuous and real-time collection of information from social media, TORCIA can detect situations of potential emergency and identify their geographical position. The TORCIA platform supports emergency operators from different organizations and institutions with a decision-support dashboard and favors the creation of a collaborative process that combines the contributions of citizens and institutions by means of a mobile app that is also integrated within the platform.*

**Keywords:** Emergency management, Crisis management, Social media, Twitter, Semantic engine

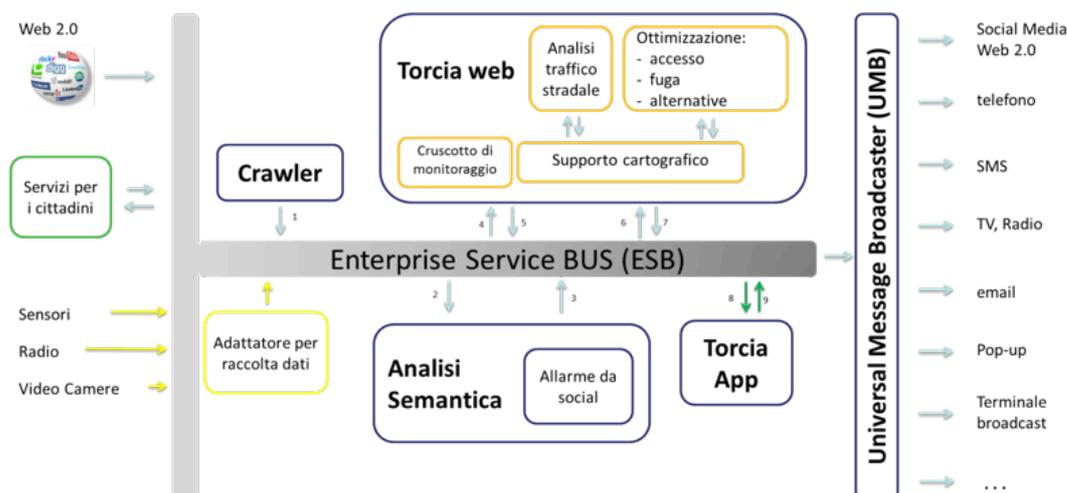


## 1. Introduzione

TORCIA è un progetto finanziato da Regione Lombardia per lo sviluppo di una piattaforma che valorizzi l'utilizzo dei social media nei processi di gestione delle emergenze e, più in generale, nei processi di pianificazione e controllo delle situazioni critiche - incidenti o calamità naturali - garantendo la resilienza delle infrastrutture di trasporto sia per quanto riguarda l'infrastruttura di telecomunicazione sia quella della rete stradale. L'elaborazione del progetto ha avuto inizio nel giugno 2012 e si è conclusa nel giugno 2014. La partnership del progetto è composta da Alcatel-Lucent, capofila di TORCIA, dal Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB) del Politecnico di Milano, che ha la responsabilità scientifica del progetto, da Fondazione Politecnico di Milano, e dalle società ACT Solutions, Beta 80 e Vidiemme.

I Comuni italiani, sulla base delle proprie realtà e criticità territoriali, pianificano la risposta alle possibili emergenze tramite i piani comunali di protezione civile, che prevedono una serie di operazioni, ruoli e misure da adottare al verificarsi di una determinata calamità naturale o antropica. La viabilità è uno dei punti chiave da pianificare in caso di emergenza: il piano comunale di protezione civile infatti può prevedere la modifica del senso di circolazione e lo sgombero di determinate arterie per consentirne l'utilizzo ai soli mezzi di soccorso. Mantenere le condizioni di sicurezza e indicare la percorribilità dell'infrastruttura stradale sono elementi chiave per una corretta gestione delle situazioni di emergenza.

In questo contesto, il progetto TORCIA mira a facilitare le operazioni di raccolta dei dati sul territorio per consentire un'immediata comprensione delle situazioni critiche e delle iniziative da compiere. I cittadini utenti del Web 2.0 diventano protagonisti nel fornire, tramite i social network - facebook e twitter in primis - informazioni georeferenziate e dati utili per una migliore gestione della criticità in atto. Per raccogliere, razionalizzare e interpretare queste informazioni e condividerle con le autorità competenti, il progetto TORCIA ha messo a punto una infrastruttura di elaborazione basata su piattaforme cloud geograficamente distribuite e collegate tra loro da una rete ottica ad alta capacità. In questo modo TORCIA è in grado di sfruttare la collaborazione della cittadinanza analizzando i dati condivisi, utilizzando in modo tecnologicamente innovativo, integrato e completo la rete dei social network, sia per interpretare le informazioni in fase di emergenza, sia per fornirle alle autorità. La Figura 1 mostra l'architettura tecnologica complessiva del progetto Torcia.



**Figura 1**  
L'architettura della piattaforma TORCIA

## 2. L'informazione sociale

Il workflow di TORCIA parte con la raccolta continua e in tempo reale di informazioni dai principali social media. La prima domanda che occorre porsi è se i social media offrano effettivamente informazione pertinente al contesto del quale ci si sta occupando e utile per il raggiungimento degli obiettivi che ci si è posti. Nel nostro caso, è possibile rispondere positivamente a questa domanda se gli utenti dei social media, ovvero i cittadini, parlano di situazioni di emergenza sui social media e se tale informazione può essere effettivamente utile ai fini della gestione dell'emergenza stessa. La verifica di queste condizioni è stata oggetto di una interessante analisi preliminare di fattibilità che è stata condotta all'inizio del progetto.

La prima, ovvia, considerazione da fare è che se un'emergenza si verifica in un luogo scarsamente popolato, è molto probabile che non ci sia un numero sufficiente di utenti social media fra le persone coinvolte e che, di conseguenza, i social media siano poco informativi. Ad esempio, una frana di un sentiero a bassa percorrenza riceverà poca o nessuna attenzione sui social media da parte delle pochissime persone coinvolte. Viceversa, una frana di un sentiero famoso (ad esempio la frana della Via dell'Amore alle Cinque Terre) sarà tempestivamente segnalata sui social media. Generalizzando, i social media diventano tanto più utili quanto maggiore è il numero di persone coinvolte. Ci siamo dunque concentrati sul caso più favorevole, quello delle *emergenze cittadine* e, in particolare, delle *alluvioni*, divenute tristemente frequenti e perciò adatte a fornire numerosi test case in corso di progetto.

Con riferimento alle alluvioni in un contesto cittadino, le nostre analisi preliminari si sono poste i seguenti obiettivi:

- verificare i volumi di parlato su Twitter relativo alle alluvioni;
- verificare gli argomenti del parlato su Twitter relativo alle alluvioni;
- verificare la presenza di elementi utili alla geolocalizzazione del parlato in caso di alluvioni;
- verificare la presenza e il ruolo delle istituzioni su Twitter;
- confrontare la situazione italiana con le best practice internazionali.

ALLAGAMENTO	7	INONDATA	41
ALLAGAMENTI	58	INONDANDO	1
GRANDINE	74	ALLAGARE	29
GRANDINATA	12	ALLAGATO	40
GRANDINATE	2	ALLAGATA	79
TEMPESTA	337	ALLAGANDO	16
ALLUVIONE	30	ESONDARE	1
INONDAZIONE	3	ESONDATO	1
INONDAZIONI	31	ESONDANDO	0
ESONDAZIONE	1	FRANARE	7
FRANA	721	FRANATO	2
FRANE	52	FRANATA	10
URAGANO	69	FRANANDO	0
NUBIFRAGI	15	ALLUVIONATO	0
NUBIFRAGIO	47	ALLUVIONATE	0
INONDARE	4	ALLUVIONATI	6
INONDATO	20	ALLUVIONATA	1
		GRANDINE	8

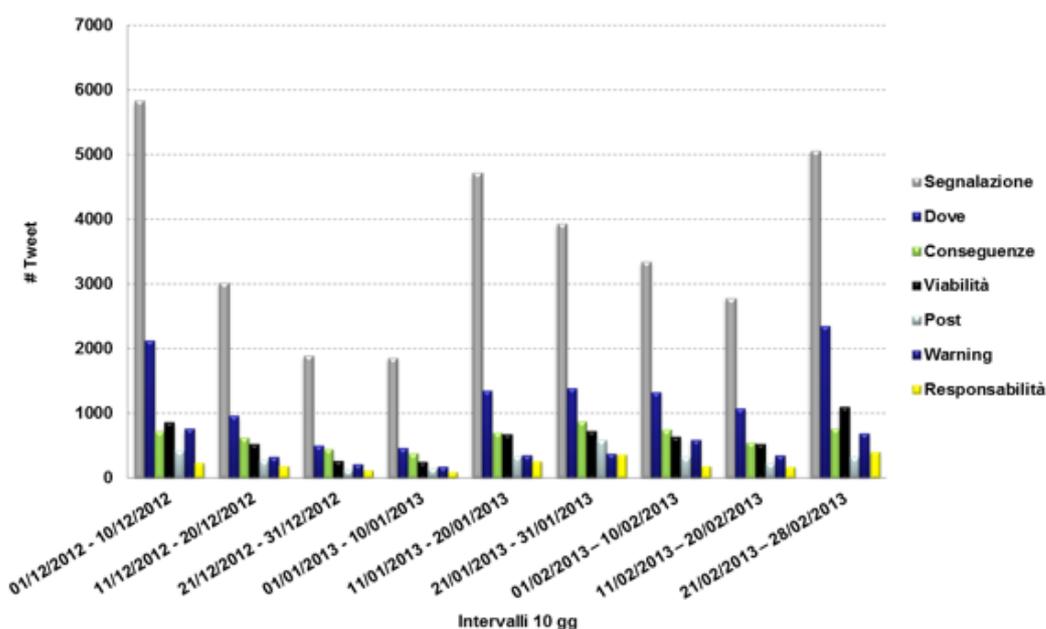
**Figura 2**

*Parole chiave per la raccolta di informazioni da Twitter (crawling keyword).*

Il crawling, ovvero la raccolta di post, è stato attivato nel settembre 2012 su circa 60 keyword che definiscono il dominio alluvioni (in Italiano e in Inglese, fra cui allagamento, frana, alluvione, tempesta, uragano, inondazione, nubifragio, hurricane, flood, eccetera). La lista delle keyword in Italiano è riportata a titolo esemplificativo nella Figura 2. Complessivamente, sono stati analizzati oltre 40 milioni di tweet nel corso del progetto. I volumi riportati nella Figura 2 rappresentano il numero di occorrenze in una giornata di maltempo tipica (priva di situazioni che si possano definire di emergenza). Si nota come il parlato tenda a concentrarsi su alcune parole chiave, che, tuttavia, tendono a cambiare nel tempo a seconda sia delle condizioni di maltempo, sia dell'effetto di imitazione nella scelta delle parole che si crea nel momento in cui gli utenti si scambiano opinioni sul maltempo. Si può concludere che il parlato in lingua italiana sul dominio delle alluvioni ha volumi considerevoli (in media 40.000 tweet/mese) e può essere oggetto di ulteriori analisi (vedi anche [4]). Il volume di parlato in lingua Inglese è di un ordine di grandezza superiore [5].

Abbiamo poi ricavato *bottom-up* dall'analisi del parlato gli argomenti di conversazione più frequenti [3]. Il nostro obiettivo in questo caso era capire se il parlato può offrire indicazioni potenzialmente interessanti nella gestione delle emergenze. Le categorie di parlato più frequenti si sono rivelate le seguenti:

- *Dove*, ovvero indicazione di luoghi nei quali si sta verificando una situazione di maltempo.
- *Segnalazione*, ovvero indicazione di situazioni di maltempo e valutazione della loro gravità (da semplice disagio a vero e proprio allarme).
- *Conseguenze*, ovvero indicazione di fatti rilevanti accaduti come conseguenza del maltempo (allagamenti, crolli, frane, eccetera).
- *Responsabilità*, ovvero opinioni sull'attribuzione della responsabilità delle conseguenze del maltempo.
- *Viabilità*, ovvero conseguenze del maltempo sul traffico e sulla viabilità.
- *Warning*, ovvero discussione preliminare al maltempo di previsioni meteorologiche negative.
- *Post-emergenza*, ovvero discussione di eventi da poco trascorsi che hanno creato situazioni di emergenza, spesso in relazione alla gestione delle attività di ripristino della normalità.



**Figura 3**  
*Gli argomenti delle conversazioni online (dominio alluvioni).*

La Figura 3 riporta la distribuzione del parlato sugli argomenti di conversazioni sopra descritti. Si nota una prevalenza di attenzione per la categoria *segnalazione* che, a sua volta, indica un ruolo dominante del parlato durante l'emergenza, piuttosto che nelle fasi che precedono e che seguono l'emergenza. Il fatto che le fasi di pre- e post-emergenza siano le meno discusse è parzialmente in relazione a una scarsa presenza delle autorità su Twitter, in

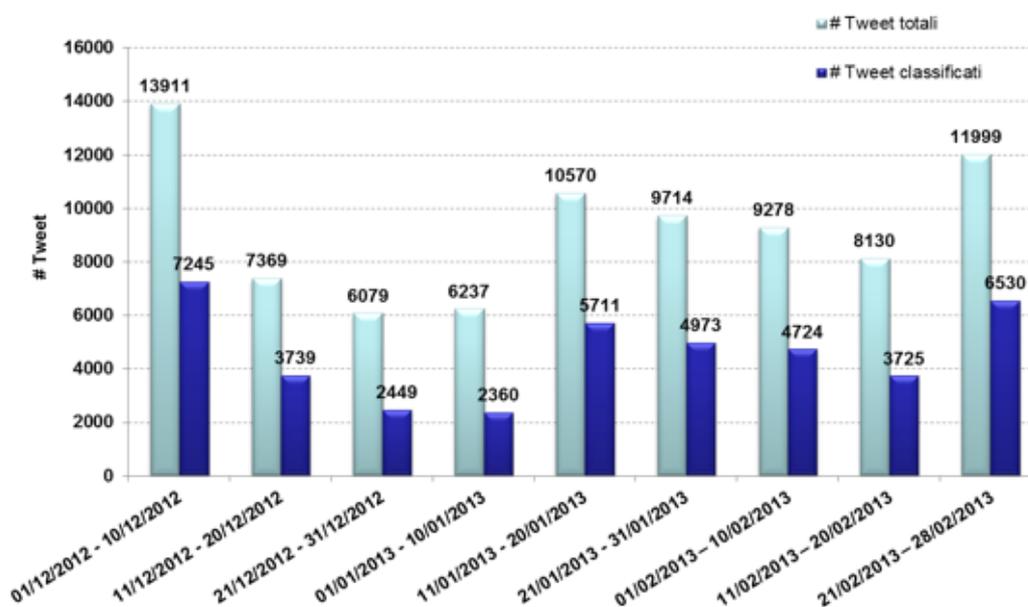
particolare in fase di pre-emergenza. Si nota inoltre un notevole volume di parlato appartenente alle categoria *dove*, ovvero post utili a definire l'area geografica interessata dall'emergenza e *viabilità*, ovvero post che forniscono indicazioni sulle strade interessate dall'emergenza.

Queste osservazioni indicano che:

- La raccolta e analisi delle informazioni da social media, Twitter in particolare, deve essere operata in tempo reale, data la prevalenza di parlato utile in fase di gestione dell'emergenza (si tratta di alcune ore).
- Le informazioni raccolte possono fornire indicazioni utili ai fini delle localizzazione dell'emergenza e, in misura minore, della identificazione di problemi di viabilità.

### 3. Lo strumento semantico

Tuttavia, non tutta l'informazione si è rivelata utile ai fini della gestione delle emergenze. Come in molti altri domini, oltre metà dei post raccolti sintatticamente tramite parole chiave di crawling si rivela non pertinente rispetto al dominio di analisi. La Figura 3 mostra come il dominio delle alluvioni non faccia eccezione, indicando come circa il 50% dei post raccolti non siano pertinenti al dominio di riferimento pur includendo una delle parole chiave di crawling riportate nella Figura 2. Dati i volumi di parlato, non è possibile pensare a una eliminazione manuale delle informazioni non pertinenti. Nell'ambito dell'architettura progetto Torcia (vedi Figura 1) è il motore semantico [2] ad occuparsi del riconoscimento e dell'eliminazione dell'informazione non rilevante.



**Figura 4**  
Volumi di parlato raccolti sintatticamente vs. volumi di parlato pertinenti al dominio delle alluvioni (esempio).

Oltre a ripulire l'informazione, il motore semantico classifica l'informazione nelle categorie di parlato precedentemente discusse. Tale classificazione è operata da un modulo software basato su una rete semantica che associa a ciascuna categoria dei metadati, pesandoli opportunamente rispetto al dominio di analisi. La selezione dei metadati e la regolazione dei pesi nella rete semantica sono stati oggetto di attenta analisi e continuo affinamento nel corso del progetto.

Il motore semantico si occupa poi di riconoscere se nell'informazione filtrata c'è l'indicazione di qualche situazione di potenziale emergenza o di emergenza in atto (nel primo caso viene creata un'*alerta*, nel secondo una *segnalazione*). L'individuazione di una situazione di potenziale emergenza, come pure quella di un'emergenza in atto avviene tramite l'individuazione di picchi nei volumi di parlato rispetto ai volumi medi. Tuttavia, si è osservato che non tutti i picchi corrispondono a una reale situazione di emergenza, potenziale o in atto e che non è quindi sufficiente fissare soglie sui volumi di parlato per creare corrispondenti allerte o segnalazioni. Ad esempio, una situazione di maltempo generalizzato su una porzione ampia del territorio nazionale genera un aumento anche molto consistente dei volumi di parlato, ma molto spesso non corrisponde a un'emergenza. Il picco nei volumi è confrontabile o addirittura superiore a quello che si è avuto durante l'alluvione a Catania del 2013 o quella in Sardegna del 2014. Ciò che permette l'individuazione di una situazione di potenziale emergenza o di emergenza in atto è la combinazione di soglie di parlato con una percentuale minima (una terza soglia) di volumi che si riferiscono a una stessa zona o località geografica (categorie *dove* e *viabilità*). Ad esempio, nel caso dell'alluvione a Catania, ciò che il motore semantico scopre analizzando i dati è un picco sopra soglia di parlato che fa contemporaneamente riferimento al dominio *alluvioni* e alla località geografica *Catania*. Il riconoscimento delle località geografiche avviene tramite interfacciamento con il servizio cartografico realizzato all'interno del progetto da Vidiemme (vedi Figura 1).

Categorie	Metadati che le compongono
Dove	Città, Centro, Via, Metro, Nord, Spagna, Venezia, Posto, Località, Zona...
Segnalazione	Allagamenti, Pioggia, Esondato, Fiume, Acqua, Tempesta, Nubifragio...
Conseguenze	Morti, Salvo, Frana, Evacuata, Inghiottiti, Situazione, Danni, Crollata...
Responsabilità	Polizia, Magistrato, Pattuglia, Autopompa, Ministri, Comune, Tilt, Capo...
Viabilità	Libera, Percorribile, Ripulita, Sgombra, Chiusa, Traffico, Percorsi, Frana...
Warning	Fogne, Intasato, Argini, Pericoloso, Previsti, Rischio, Allerta, Flusso...
Post	Risarcimento, Lamentela, Soldi, Assegni, Fondi, Denunce, Interventi...

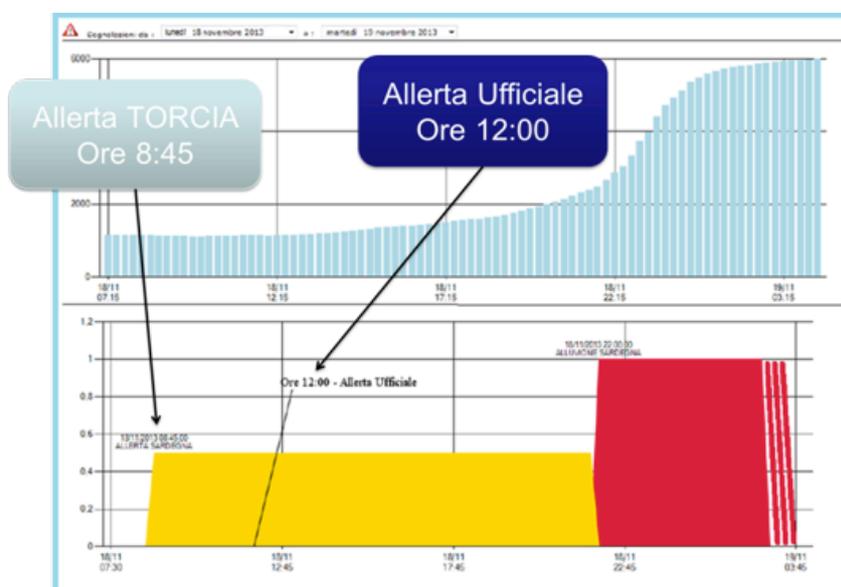
**Figura 5**  
Categorie di parlato e esempi di metadati.

## Il cruscotto direzionale

Allerte e segnalazioni vengono passate al cruscotto di monitoraggio (realizzato da Beta80, partner di progetto). Nel corso del progetto, abbiamo avuto modo di verificare l'architettura Torcia su molte situazioni di emergenza, fra cui:

- alluvione Catania Febbraio 2013,
- alluvione Vicenza Maggio 2013,
- (primo) nubifragio Roma, 07/07 - 08/07, 2013,
- (secondo) nubifragio Roma, 21/07 - 22/07, 2013,
- (terzo) nubifragio Roma, 27/08 - 28/08, 2013,
- allagamenti Toscana, 21/10 - 22/10, 2013,
- alluvione Sardegna Novembre 2013,
- alluvione Marche Aprile 2014,
- alluvione Senigallia Maggio 2014,
- esondazione del Seveso Settembre 2014.

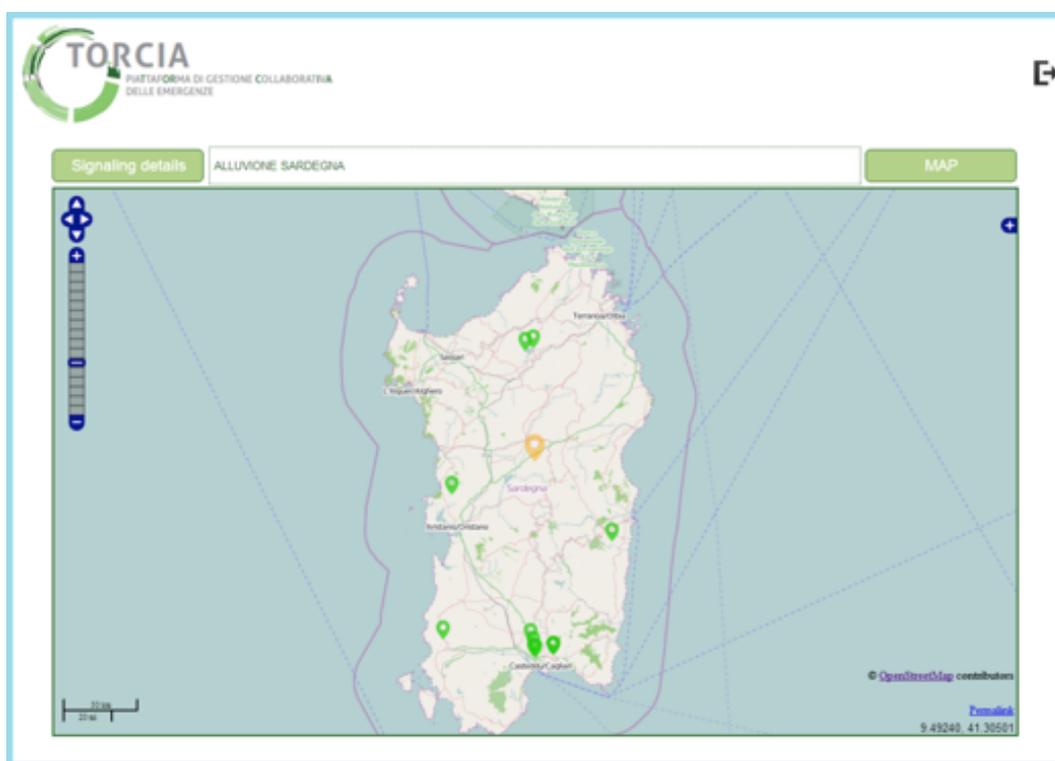
Ciò di cui ci siamo resi conto è che il parlato su social media è di per sé poco fruibile se non viene organizzato, ovvero ripulito, categorizzato e geolocalizzato. Questo sforzo di analisi permette però di ottenere informazioni interessanti a complemento di quelle provenienti da altre fonti già tradizionalmente utilizzate dagli operatori istituzionali. La Figura 6 mostra, nel caso dell'alluvione a Catania, come le allerte sollevate dal motore semantico sono inviate al cruscotto decisionale con un anticipo temporale significativo rispetto al momento dell'allerta ufficiale. Questo comportamento è consistente su tutte le emergenze analizzate.



**Figura 6**  
*I tempi delle allerte TORCIA.*

Tuttavia, l'approccio con il quale è stato progettato il cruscotto di monitoraggio è fortemente orientato all'operatore, al quale le informazioni, incluse allerte e segnalazioni, sono soltanto proposte e devono essere validate manualmente. Tale validazione è stata ritenuta necessaria sia perché il motore semantico, nonostante il continuo affinamento, resta comunque soggetto a errore, sia perché non tutta l'informazione sociale è affidabile e deve essere analizzata e validata da un operatore che ha accesso a tutte le possibili fonti informative, non solo ai social media.

La Figura 7 mostra la schermata del cruscotto con gruppi di tweet geolocalizzati nel caso dell'alluvione in Sardegna del novembre 2013. Si nota come l'informazione resa disponibile dal cruscotto sia molto più fruibile per un operatore istituzionale rispetto all'accesso diretto ai social media.



**Figura 7**

*Geolocalizzazione delle informazioni da social media: il caso dell'alluvione in Sardegna.*

#### **4. Formazione informatica e produttività degli utenti**

L'informazione organizzata e validata dagli operatori è resa disponibile ai cittadini tramite un'applicazione mobile (realizzata da Beta80). La applicazione mobile rappresenta il punto di incontro fra cittadini e operatori istituzionali, costituendo uno strumento di coordinamento e, per le istituzioni, di crowdsourcing. I cittadini possono accedere a informazione validata e,

direttamente dalla app, fornire ulteriore informazione agli operatori, inviandola direttamente al cruscotto di monitoraggio e contemporaneamente anche a Twitter. Gli operatori possono raccogliere informazioni dai cittadini, in tempo reale, e a loro volta inviare comunicazioni importanti per la gestione delle emergenze. Inoltre, tramite il meccanismo di registrazione, i cittadini utenti della mobile app rappresentano un insieme di utenti privilegiati, dei quali si può conoscere l'identità e tracciare la storia in caso di emergenze ricorrenti. E' poi possibile valutare la significatività dei contributi degli utenti della app, con i noti meccanismi di scoring degli utenti, in modo da poter operare valutazioni anche automatiche di affidabilità delle informazioni sulla base degli score.

La Figura 8 mostra l'accesso alle informazioni geolocalizzate tramite la mobile app. E' evidente come gli stessi operatori di protezione civile possano essere utenti della mobile app [1] e come, di conseguenza, la app rappresenti uno strumento di coordinamento fra cittadini e operatori istituzionali. E' possibile immaginare scenari di gestione delle emergenze in cui la applicazione mobile venga utilizzata anche per selezionare cittadini e assegnare loro un ruolo nella gestione dell'emergenza, utilizzando le informazioni dei profili utenti e la loro storia di utilizzo della app.

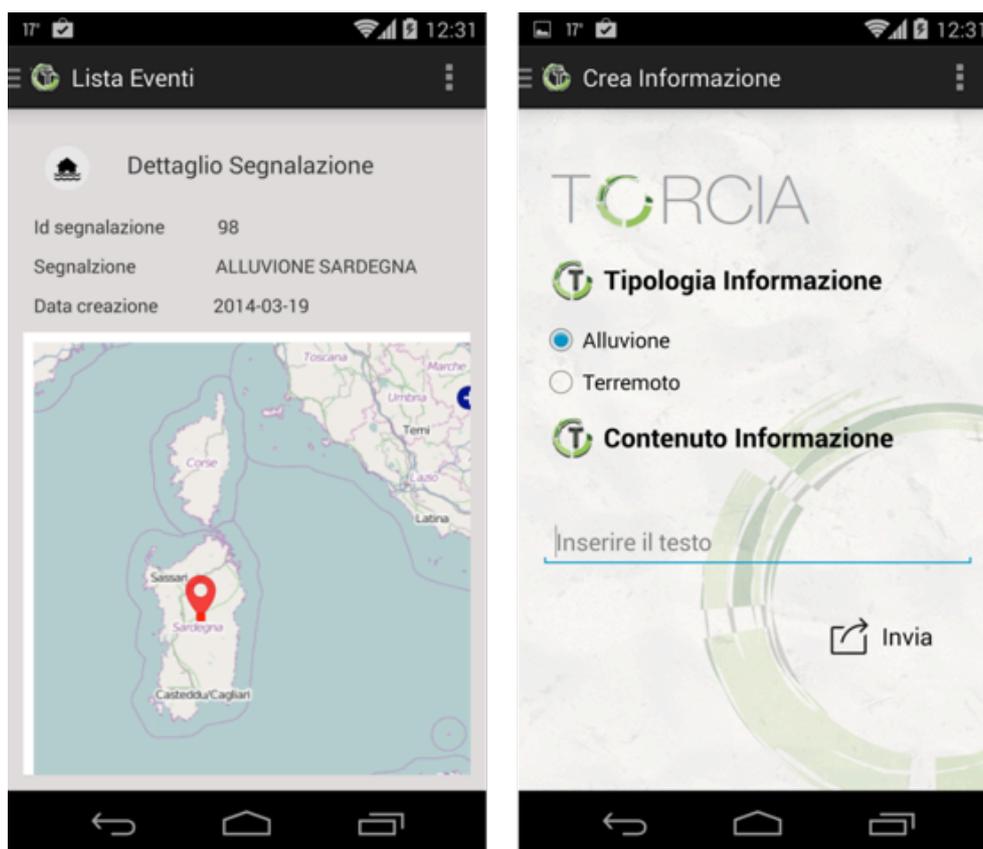


Figura 8

*Geolocalizzazione delle informazioni da social media con l'applicazione mobile.*

## 5. Conclusioni

Riassumendo, il workflow di TORCIA parte con la raccolta in tempo reale di informazioni dai principali social media. Tale informazione viene analizzata da un motore semantico che è in grado di individuare e localizzare geograficamente situazioni di potenziale emergenza in base al parlato online. A seguito di tali situazioni, viene creata un'allerta che è poi inviata ad un cruscotto di monitoraggio progettato per gli operatori istituzionali, quali la Protezione Civile. Se l'allerta viene convalidata dagli operatori, si crea una segnalazione di emergenza e si attivano tutte le procedure di gestione dell'emergenza. In particolare, all'interno della piattaforma TORCIA, è possibile comunicare con i cittadini in modalità multi-canale, includendo nello sforzo di comunicazione anche gli stessi social media. I cittadini possono utilizzare la mobile app per ottenere informazioni utili sulla situazione di emergenza. Ad esempio, possono accedere a tutta l'informazione che il motore semantico preleva dai social media visualizzandola su una mappa geografica e possono richiedere alla app il calcolo delle vie di fuga in base alla posizione corrente e alle indicazioni operative degli operatori istituzionali (questo modulo è stato sviluppato da ACT Solutions). Tramite la app possono a loro volta fornire informazioni utili per la gestione dell'emergenza, favorendo la realizzazione di un processo collaborativo che integra i contributi di cittadini e istituzioni.

## Ringraziamenti

Gli autori desiderano ringraziare le aziende partecipanti al progetto TORCIA e, in particolare, Claudio Mangone e Giorgio Parladori per il loro contributo scientifico e di coordinamento.

## Bibliografia

- [1] Capelli, P. "Una metodologia per la classificazione dei modelli competitivi delle mobile application", Tesi di Laurea in Ingegneria Informatica, Politecnico di Milano, Dicembre 2013.
- [2] Carcaci, C. "Classificazione sintattica con metodi insiemistici", Tesi di Laurea in Ingegneria Informatica, Politecnico di Milano, Dicembre 2012.
- [3] Cesana, G., D'Etto, A. "Analisi, Metodologia e Sviluppo di un Sistema Integrato per la Gestione delle Emergenze in Ambito Social", Tesi di Laurea in Ingegneria Informatica, Politecnico di Milano, Luglio 2013.
- [4] Rossitto R., "I numeri di Twitter in Italia", Wired, 26 Settembre 2012, <http://daily.wired.it/news/internet/2012/09/26/numeri-twitter-italia.html>
- [5] Lachlan, K.A., Spence, P.R., Lin, X., Del Greco, M. "Screaming into the Wind: Examining the Volume and Content of Tweets Associated with Hurricane Sandy," *Communication Studies*, 65( 5), 500-518, 2014

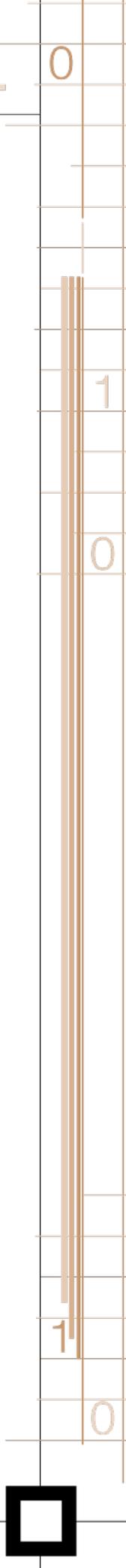
## Biografia

**Chiara Francalanci** è Professore di Sistemi Informativi del Politecnico di Milano, dove si è laureata in Ingegneria Elettronica nell'ottobre 1991. Durante gli studi di dottorato è stata ricercatrice ospite per un periodo di due anni presso Harvard Business School. Ha scritto numerosi articoli sulla progettazione dei sistemi informativi e sul valore economico dell'informatica, svolto attività di consulenza nel settore finanziario e manifatturiero, sia in Italia che negli Stati Uniti, è editor del Journal of Information Technology e senior editor delle AIS Transactions on Enterprise Systems.

E-mail: chiara.francalanci@polimi.it

**Paolo Giacomazzi** è professore di Multimedia Internet al Politecnico di Milano, dove si è laureato con lode in Ingegneria Elettronica. Ha svolto un periodo di ricerca presso la University of Mississippi, presso il National Center for Wireless Communications. Ha svolto attività di ricerca e consulenza nel settore delle reti di telecomunicazione fisse e mobili. Attualmente, si occupa di sistemi previsionali relativi sia al traffico Internet, sia ad altri domini applicativi fra i quali i social media, la grande distribuzione, e il mass market in generale.

E-mail: giacomaz@elet.polimi.it



## Rubrica

# Siamo già nel futuro

## Attività e strategie formative dell'ITS per l'ICT J. F. Kennedy di Pordenone

**Valerio Libralato**

### Sommario

L'articolo ripercorre le tappe dello sviluppo dell'Istruzione Tecnica Superiore, avviata operativamente nel 2011, attraverso l'esperienza e la vision della Fondazione Istituto Tecnico Superiore J.F. Kennedy per le Tecnologie dell'Informazione e della Comunicazione di Pordenone. Gli Istituti Tecnici Superiori (ITS) nascono come risposta alla costante richiesta di tecnici qualificati proveniente dal sistema produttivo italiano - dalle PMI in particolare - e rappresentano, per i diplomati della scuola superiore, una nuova opportunità oltre ai percorsi accademici e all'ingresso immediato nel mondo nel lavoro. La stretta collaborazione con le realtà economiche presenti sul territorio e un curriculum didattico ampiamente dedicato all'esperienza pratica, fanno sì che la capacità di inserimento occupazionale risulti essere il tratto distintivo, auspicato e realizzato, degli ITS.

La Fondazione ITS Kennedy opera nell'ambito della figura dello sviluppatore software. I punti di forza risiedono nella sinergia con il tessuto locale delle aziende ICT per la definizione dei profili e la progettazione degli obiettivi didattici; nell'utilizzo quasi esclusivo di docenti di provenienza ed esperienza aziendale; nell'orientamento verso tecnologie allo stato dell'arte, cloud e mobile in particolare; nell'utilizzo dello standard europeo dell'e-Competence Framework, quale riferimento per la descrizione e lo sviluppo delle competenze. Gli indicatori di risultati sono rappresentati da una nutrita rete di aziende che collaborano attivamente per la realizzazione delle attività - 98 aziende italiane e 11 in cinque paesi europei - e da esiti occupazionali particolarmente lusinghieri - oltre il 90% dei 42 diplomanti occupati con mansioni coerenti al proprio profilo.



Le strategie evolutive della Fondazione ITS Kennedy si concentrano su un coinvolgimento proattivo delle aziende del territorio, sul consolidamento della possibilità di effettuare lo stage all'estero, sulla realizzazione di percorsi ITS in Veneto, sull'estensione della durata dei corsi a tre anni, a partire dall'anno scolastico 2016-17.

## Abstract

The article describes the development steps of the High Level Technical Education (Istituzione Tecnica Superiore), which was effectively started on 2011, through the experience and the vision of the Fondazione Istituto Tecnico Superiore J.F. Kennedy for Information and Communication Technologies in Pordenone. The High Level Technical Institutes (Istituti Tecnici Superiori - ITS) are born as a response to the constant demand for qualified technicians from the Italian production system - SMEs in particular - and, for high school graduates, represent a new opportunity in addition to the academic courses and to the immediate entrance in the labour market. The close cooperation with the economic realities in the area and a curriculum largely devoted to practical experience, enable the capacity of job placement to be the advocated and implemented hallmark of the ITS.

The Fondazione ITS Kennedy works within the software developer profile. The strengths lie in the synergy with the local ICT companies for profiling and designing of educational goals; in the almost exclusive use of teachers coming from companies; in the orientation to the state of the art technologies, cloud and mobile in particular; in the use of standard European e-Competence Framework, as a reference to the description and skills development. The performance indicators are represented by a large network of companies which actively cooperate for the activities realization - 98 Italian companies and 11 in five European countries - and particularly flattering employment outcomes - more than 90% of 42 graduating employees with tasks consistent with their profiles.

The development strategies of the Fondazione ITS Kennedy focus on a proactive involvement of the local companies, on the making of the internship abroad likely realizable, on the implementation of ITS courses in Veneto, on extending the duration of the course of three years, for the school year 2016-17.

**Keywords:** High Level Technical Education, post-secondary education, European e-Competence Framework, internship, learning-by-doing, labour market, ICT industry, employment opportunities, local ICT companies

## 1. La formazione ITS

I percorsi formativi post-secondari dell'Istruzione Tecnica Superiore, della durata di quattro o sei semestri, si inseriscono nell'architettura del sistema di istruzione e formazione italiano dopo la scuola secondaria di secondo grado parallelamente ai percorsi universitari. Essi si caratterizzano per la convinta integrazione tra scuola e lavoro, secondo la modalità formativa del "sistema duale"<sup>1</sup> già da tempo applicata in diversi paesi europei.

Dopo l'esperienza dell'apprendistato e dei percorsi di Istruzione e Formazione Professionale (IeFP) [1] – nei quali i preadolescenti italiani possono completare l'obbligo scolastico – l'ITS rappresenta per il nostro paese il primo tentativo strutturato e istituzionalizzato di completamento del sistema duale nel segmento terziario dell'istruzione pubblica, successivo al diploma di scuola secondaria superiore.

I corsi ITS si caratterizzano per l'importanza assegnata al sistema delle imprese e del lavoro come opportunità formativa: le Fondazioni alle quali è affidata la gestione degli ITS devono prevedere una significativa presenza di imprese al loro interno; la formazione d'aula deve integrarsi con tirocini lavorativi (almeno 30% del monte ore) attraverso i quali le competenze acquisite possano trovare effettiva concretizzazione e verifica operativa; la docenza dev'essere affidata per almeno la metà a professionisti provenienti dal sistema produttivo.

In altre parole i percorsi ITS si contraddistinguono per un forte orientamento al conseguimento di competenze professionali facilmente riconoscibili e spendibili sul mercato del lavoro, in un contesto di obiettivi a medio termine che mirano a:

- contribuire alla diffusione della cultura tecnica e scientifica per sostenere lo sviluppo economico e la competitività del sistema produttivo italiano;
- rispondere alla crescente richiesta del mercato del lavoro di tecnici superiori, in grado di coniugare conoscenze culturali e formazione tecnica e professionale, proveniente in particolare dalle PMI e dalle imprese orientate all'internazionalizzazione e all'innovazione tecnologica;
- rafforzare la collaborazione con il territorio, il mondo del lavoro, le sedi della ricerca scientifica e tecnologica, il sistema della formazione professionale.

L'Istruzione Tecnica Superiore prende avvio con il D.P.C.M. 25 gennaio 2008 "Linee Guida per la riorganizzazione del sistema di Istruzione e Formazione Tecnica Superiore e la costituzione degli Istituti tecnici superiori"[2], che ne ha definito l'architettura progettuale e didattica e le caratteristiche generali in termini di certificazioni, posizionamento ed integrazione con i sistemi di istruzione e formazione a livello post-secondario e con gli standard europei in materia di riconoscimento delle competenze (Livello 5 EQF - European

<sup>1</sup> Sistema di alternanza scuola lavoro, sviluppato in particolare in Svizzera e Germania, nel quale coesistono parallelamente percorsi di istruzione in senso stretto e percorsi integrati di istruzione e lavoro

Qualification Framework)<sup>2</sup>[3][4][5]. Successivamente il D.M. MIUR 7 settembre 2011 [6], attraverso la definizione delle aree e ambiti settoriali, delle figure nazionali di riferimento e dei relativi standard di competenza, nonché delle modalità per la verifica finale delle competenze acquisite e della certificazione delle stesse, ha di fatto dato il via all'erogazione della formazione ITS. I primi percorsi, avviati nel 2011, si sono conclusi nell'estate del 2013 con il rilascio dei primi diplomi di Tecnico Superiore.

A suggellarne anche esteriormente il loro essere titoli di studio rilasciati dalla Repubblica italiana, i diplomi dei tecnici superiori sono redatti sul pregiato supporto cartaceo fornito dalla zecca dello Stato con numerazione progressiva che, proprio per i primi diplomati del 2013, ha quindi annoverato il diploma numero uno del primo tecnico Superiore.

## 2. La Fondazione ITS Kennedy

La Fondazione ITS Kennedy per le Tecnologie dell'Informazione e della Comunicazione è nata a Pordenone nel settembre 2010 su impulso della Regione Friuli Venezia Giulia e per volontà del sistema della formazione, degli Enti locali e delle aziende. Vi aderiscono:

- per le associazioni di categoria: Unione Industriali della Provincia di Pordenone - cui è assegnata la presidenza della Fondazione - e AICA, Associazione Italiana per l'Informatica e il Calcolo Automatico;
- per i soggetti formativi, enti di consulenza e di ricerca: Istituto Tecnico Settore Tecnologico J.F. Kennedy, quale Istituto Tecnico di riferimento; ITT G. Chilesotti di Thiene (VI); Comitato regionale dell'ENFAP del Friuli Venezia Giulia; Consorzio Universitario di Pordenone; Consorzio Friuli Formazione; E.N.A.I.P. Friuli Venezia Giulia; Fondazione Opera Sacra Famiglia; IAL Innovazione Apprendimento Lavoro FVG S.R.L. Impresa Sociale; I.R.E.S. - Istituto di Ricerche Economiche e Sociali del FVG; Agenzia per lo sviluppo economico della montagna Agemont Spa; Polo Tecnologico di Pordenone Andrea Galvani;
- per le aziende: Eurotech Spa; INSIEL Spa; Gruppo Servizi CGN; Real Comm Srl; Space Computer Srl; Springfirm Srl; Tempestive Srl; Teorema Engineering Srl; Umana Spa; VDA Elettronica Spa; Banca di Credito Cooperativo del Pordenonese;
- per le amministrazioni pubbliche: Comune di Pordenone, Provincia di Pordenone.

La Fondazione Kennedy è una delle sei che, sull'intero territorio nazionale, opera nell'ambito "6.1 - Metodi e tecnologie per lo sviluppo dei sistemi software".

Le figure professionali TS da formare vengono individuate dalla Fondazione come specificazioni del contesto locale e delle specializzazioni tecnologiche

<sup>2</sup> L'European Qualification Framework - Quadro europeo delle qualifiche per l'apprendimento permanente (EQF) è un quadro comune di riferimento che collega fra loro i sistemi di qualificazione dei Paesi dell'Unione Europea, fungendo da dispositivo di traduzione utile a rendere le qualifiche omogenee tra paesi e sistemi formativi differenti

alla luce degli attuali e più recenti trend di settore. La progettazione non si limita pertanto a declinare a livello locale il profilo nazionale, in risposta ai bisogni più immediati del territorio, ma si propone di preparare una figura professionale adeguata ad inserirsi nel mercato ICT nell'arco dei due anni e mezzo che separano la progettazione dall'inserimento lavorativo dei diplomati. Un tempo molto lungo, in considerazione dei ritmi di sviluppo e crescita tecnologica dell'ICT, che richiede da parte degli organi decisionali della Fondazione e del gruppo di progetto una capacità di visione almeno altrettanto lunga.

Seguendo tale logica e secondo le indicazioni delle aziende partner, fin dal 2011, il cloud e il mobile sono stati identificati quali ambienti di riferimento in termini infrastrutturali e tecnologici. I profili proposti da allora si configurano come una specializzazione finalizzata al raggiungimento delle competenze proprie della figura di riferimento, attraverso l'utilizzo delle tecnologie e degli approcci di logica e fruibilità dei servizi, che rappresentano la tendenza evolutiva del settore.

### 3. Le attività sviluppate

L'esperienza e le competenze della Fondazione discendono, da un lato, dalle attività direttamente promosse e gestite dalla stessa e, dall'altro, dall'esperienza pregressa dei partner che la costituiscono, ciascuno con particolare riferimento alle proprie finalità e specificità operative, sulle quali per l'appunto si fondano i motivi dell'adesione alla Fondazione ed il contributo che ciascun partner apporta.

Per quanto concerne la Fondazione, l'attività formativa è iniziata nel 2011. Da allora sono stati avviati sei corsi biennali della durata di 2000 ore ciascuno, altri due sono in fase di progettazione e promozione e saranno avviati nel prossimo autunno. Tutti i corsi fanno riferimento al profilo di "Tecnico Superiore per i metodi e le tecnologie dello sviluppo di sistemi software", con varie declinazioni:

- 2011-2013: "Tecnico Superiore per le Applicazioni Distribuite", concluso nel luglio 2013 con la qualificazione di 23 tecnici superiori, 22 di essi, un anno dopo il corso, hanno trovato lavoro con compiti coerenti al percorso;
- 2012-2014: "Tecnico Superiore per le Applicazioni Integrate su dispositivi Mobili", concluso nel luglio 2014 con la qualificazione di 19 tecnici superiori, 15 di essi, sei mesi dopo il corso, hanno trovato lavoro con compiti coerenti al percorso;
- 2013-2015: "Tecnico Superiore per le Applicazioni Integrate su dispositivi Mobili" e "Tecnico Superiore per le Applicazioni di Data Integration in ambiente Cloud", frequentati rispettivamente da 21 e 17 allievi, al secondo anno di corso;
- 2014-2016: "Tecnico Superiore per le Architetture ed i servizi Cloud" e "Tecnico Superiore per l'Intelligent Data Analysis", frequentati rispettivamente da 21 e 24 allievi, al primo anno di corso;

- 2015-2017: "Tecnico Superiore per le Architetture ed i servizi Cloud" e "Tecnico Superiore per lo sviluppo su Mobile Device", in avvio al prossimo autunno.

Tutte le attività didattiche, la cui frequenza è obbligatoria, sono realizzate presso il campus universitario del Consorzio Universitario di Pordenone (socio della Fondazione) che mette a disposizione le strutture logistiche, le attrezzature tecnologiche ed i servizi residenziali e di ristorazione determinanti per la frequenza degli studenti fuori sede. In materia di diritto allo studio la Regione Autonoma Friuli Venezia Giulia riconosce agli studenti dei corsi ITS i medesimi benefici di cui godono gli studenti universitari, tra questi l'accesso alle misure di sostegno economico erogate dall'A.R.D.I.S.S (Agenzia Regionale per il Diritto agli Studi Superiori) in relazione ai criteri di merito e reddito.

Un ruolo importante è affidato al partner IALFVG (IAL Innovazione Apprendimento Lavoro FVG S.R.L. Impresa Sociale) che, alla luce di una lunga esperienza nella formazione post diploma per il settore ICT ed in stretta collaborazione con la Fondazione stessa, l'Istituto tecnico di riferimento, il Consorzio universitario di Pordenone ed altri partners, condivide le risorse umane per la direzione, la progettazione, il coordinamento, ed il monitoraggio dei corsi.

Oltre alle attività corsuali di Tecnico Superiore, la Fondazione Kennedy promuove e realizza diverse attività sia formative che non, nell'ambito delle Azioni di sistema che si estrinsecano in tre filoni di lavoro:

- rafforzamento delle competenze ICT, soprattutto di tipo professionale, degli studenti degli ultimi anni degli istituti di formazione tecnica secondaria - potenziale bacino d'utenza per la futura offerta ITS - strutturata come attività corsuali tra le 20 e le 40 ore;
- attività promozionali e divulgative, per la promozione delle competenze e professioni ICT, sempre per gli studenti delle scuole superiori, realizzata in forma di seminari, eventi dimostrativi di tecnologie e visite aziendali;
- attività di supporto alla progettazione e gestione operativa, riguardanti la messa a punto di dispositivi di valutazione degli apprendimenti e di certificazione delle competenze in esito dai corsi ITS e la referenziazione delle stesse secondo standard nazionali ed internazionali.

#### 4. L'architettura didattica dei corsi

Il modello didattico della formazione ITS ha una forte componente esperienziale (learning-by-doing), direttamente finalizzato a sostenere un rapporto diretto e privilegiato con il mondo del lavoro. L'elemento più evidente ed immediato in tal senso risiede in un ampio ricorso alla modalità didattica del tirocinio in azienda (che può incidere fino ad 40% del monte ore di formazione). Altrettanto determinanti ed innovativi sono, sul piano operativo, il ricorso a docenti proveniente per almeno il 50% dal mondo della aziende, e, sul piano metodologico, una struttura di obiettivi formativi centrati su Unità di Competenza (UC) a loro volta declinate dalle macrocompetenze specifiche di ciascuna figura di Tecnico Superiore.

La Fondazione Kennedy, all'interno di tale quadro di riferimento, ha sviluppato un proprio modello didattico basato sulle seguenti strategie progettuali e realizzative:

- Proposta di un'offerta formativa su tecnologie allo stato dell'arte; in primo luogo perché, in ICT, ciò che costituisce innovazione oggi sarà d'uso comune in una manciata di mesi, secondariamente per far sì che i giovani Tecnici Superiori siano essi stessi veicolo di innovazione nelle realtà in cui si collocheranno.
- Coinvolgimento delle aziende ICT del territorio - a partire da quelle partner della Fondazione, per proseguire successivamente con quelle in cui sono stati realizzati gli stage aziendali - nella progettazione delle attività formative, sia in fase di definizione del profilo professionale oggetto della formazione, che di architettura delle unità formative che, infine, di obiettivi e contenuti delle stesse.
- Strutturazione del percorso in unità formative coerenti e correlate alla struttura di competenze, sviluppate secondo logiche e livelli di complessità e multidisciplinarietà crescenti: Unità Formative (UF) Base, UF Tecnico-Professionali di Base, UF Tecnico-Professionali Avanzate.
- Finalizzazione delle UF di Base alla costruzione di una "cultura aziendale digitale" che rappresenta il completamento delle competenze tipicamente tecnologiche del profilo professionale, al fine di un efficace inserimento nel mercato del lavoro.
- Ampia prevalenza, nella formazione d'aula, esercitazioni pratiche, simulazioni e realizzazione di project work, rispetto alla trasmissione di contenuti e nozioni teoriche.
- Realizzazione di un project work nel quale vengono messe a frutto le competenze e le esperienze acquisite in particolare nelle UF Tecnico-Professionali Avanzate e durante lo stage.
- Massimizzazione delle ore di stage rispetto a quanto previsto dal quadro normativo: 320 ore al primo anno e 480 al secondo, per un totale di 800, rispetto ad un monte ore complessivo del corso pari a 2000 ore.
- Utilizzo sempre più intenso di provenienti dalle aziende passando dalle competenze di base a quelle tecnico-professionali, fino al loro impiego in via praticamente esclusiva per le competenze ICT più avanzate
- Implementazione di laboratori informatici costituiti da pc portatili - assegnati in comodato d'uso agli studenti - e da adeguate infrastrutture tecnologiche di connettività, server fisici e virtualizzati e ambienti di sviluppo e svolgimento di tutte le lezioni d'aula all'interno dei laboratori.

## 5. Le competenze di riferimento

Le competenze di riferimento per i percorsi di Istruzione Tecnica Superiore sono definite dall'Allegato F del già citato D.M. MIUR 7 settembre 2011, a livello di ciascuna area tecnologica e figura professionale. Trattandosi tuttavia di macrocompetenze, tale quadro va declinato in una struttura di competenze più

direttamente verificabili e traducibili in obiettivi formativi. Nei primi due progetti sviluppati dalla Fondazione, tale declinazione è avvenuta riadattando e integrando ad hoc le competenze già definite per le corrispondenti figure dei percorsi IFTS (Istruzione e Formazione Tecnica Superiore).

In una seconda fase, cogliendo le indicazioni del Comitato Tecnico Scientifico, è stato avviato un processo di standardizzazione delle competenze, assumendo come riferimento i profili EUCIP (European Certification of Informatics Professionals) [7]. Una prima analisi, realizzata sia in fase di progettazione dei corsi che all'interno di uno studio dedicato alla realizzazione di un sistema di valutazione delle competenze, ha portato ad adottare il syllabus del modulo EUCIP Core "A - Plan: Uso e gestione dei sistemi informativi" quale struttura di contenuti per l'UF Base "I servizi informativi a supporto dei processi aziendali", comune a tutti i corsi. Successivamente, anche a fronte della necessità di uniformare ed omogeneizzare le conoscenze ICT degli studenti - la cui provenienza dai percorsi liceali è in crescita - si è scelto strutturare i percorsi:

- in un primo anno sostanzialmente comune a tutti i percorsi, dedicato all'acquisizione di un insieme di competenze ICT professionali di base;
- in un secondo anno di specializzazione, in cui affrontare le competenze caratterizzanti ciascun profilo.

All'interno di tale architettura il riferimento per le competenze ICT di base è stato individuato nell'EUCIP Core, con i moduli "B - Build: Sviluppo e implementazione di sistemi informativi" e "C - Run: Esercizio e supporto di sistemi informativi". Dal punto di vista delle competenze gestionali e organizzative, oltre al modulo Plan dell'EUCIP, sono stati adottati i syllabus delle certificazioni "Trattamento dei dati personali: Privacy e sicurezza", "European Project Management Qualification e-PMQ Fundamentals"[8] e "Patente Europea per la Qualità - EQDL start"[9].

### 5.1. Il framework e-CF

Su un piano più generale il processo di convergenza verso standard di competenza si è infine orientato, secondo una linea più chiara e definitiva, verso l'European e-Competence Framework 3.0 [10]. La struttura e-CF secondo competenze, solo successivamente convergenti in profili, le modalità di descrizione in quattro dimensioni delle competenze stesse e, infine, la disponibilità e trasparenza della metodologia di definizione e adattamento delle competenze stesse, si sono rivelate elementi decisamente interessanti ed accattivanti ai fini dell'avvicinamento a questo standard. Tale scelta rende peraltro evidenti aspetti di complessità non trascurabili, prima fra tutti la necessità di declinare e adattare la struttura di competenze per ciascuna singola figura di Tecnico Superiore. Tali rischi, tuttavia, dovrebbero venire compensati da un più facile ed immediato confronto con le aziende tanto nella fase di progettazione che in quella di gestione: il "linguaggio delle competenze", familiare al sistema della formazione, si dimostra di difficile decodifica per il mondo delle aziende. La descrizione delle competenze in termini di conoscenze e abilità e la graduazione dei livelli di possesso, con particolare riferimento all'autonomia, paiono possedere maggiori caratteristiche di interscambio e riconoscibilità tra i due universi semantici.

Dal punto di vista operativo, la scelta dello standard e-CF è stata inizialmente percorsa curvando e adattando il profilo e-CF "Developer"[11] secondo una logica di percorribilità (in termini di livelli di competenza raggiungibili) e di pertinenza (in termini di contenuti/elementi di competenza) in relazione alle caratteristiche degli specifici obiettivi formativi di ciascun corso. Identificata l' "infrastruttura" di competenze e-CF di riferimento per il corso, l'allineamento è stato condotto a livello di obiettivi di Unità Formative, ovvero senza scendere alla declinazione e identificazione di knowledges e skills caratteristiche di ciascuna e-Competence: il livello di descrizione di tali elementi è infatti volutamente "alto" e "generico" rispetto a quanto richiesto per lo sviluppo di un progetto formativo. Si tratta tuttavia solo di uno slittamento temporale di tale obiettivo, che, per essere efficacemente portato a termine, richiede tempi e percorsi un po' più articolati di quelli gestibili nell'orizzonte temporale della progettazione di un singolo corso.

Il passaggio logicamente conseguente nel percorso di adozione delle e-CF, ovvero la curvatura delle macrocompetenze ITS secondo la struttura delle e-CF, non è invece stato realizzato. Tale rinuncia è stata determinata da ragioni di tipo metodologico, sostanzialmente per la diversa natura di definizione delle due strutture. La prima – macrocompetenze ITS – sviluppata secondo fasi di processo orizzontali che non identificano propriamente competenze, ma piuttosto fasi o compiti di lavoro, senza una differenziazione in termini di profondità. La seconda – e-CF – basata su domini di competenza di tipo specialistico aggregati in aree, con graduazione verticale in proficiency levels (livelli di competenza espressi con riferimento ai livelli EQF), che riesce invece ad individuare competenze (in senso stretto), valide e spendibili attraverso diverse fasi del processo produttivo.

La svolta risolutiva sul fronte dell'e-CF si è nel frattempo manifestata con il D.Lgs. 16 gennaio 2013, n. 13 [12] e la prevista creazione del "Repertorio nazionale di titoli di istruzione e formazione e delle qualificazioni professionali", nel cui ambito di sviluppo il Gruppo tecnico di Regioni e P.A. [8], ha messo a punto un Protocollo metodologico per la costruzione del citato Repertorio nazionale. In tale quadro, in Friuli Venezia Giulia lo standard e-CF è stato adottato per la definizione delle Qualificazioni Professionali Regionali (QPR) per il settore ICT. Il repertorio delle QPR per il settore dei "Servizi di informatica" è in fase di realizzazione con la collaborazione di AICA (Associazione Italiana per l'Informatica e il Calcolo Automatico) e se ne prevede la pubblicazione di una prima versione nel maggio del corrente anno (2015).

Il format delle QPR prevede la descrizione della QPR stessa e la sua declinazione in conoscenze e abilità per ogni specifico livello EQF, così come la costruzione di Schede di Situazioni Tipo (SST) che descrivono un quadro di "output" (o "deliverables", nella terminologia e-CF) da utilizzare quali parametri per la verifica del possesso della competenza. Tanto le QPR quanto le SST potranno essere direttamente utilizzate nella progettazione e nelle fasi di valutazione dei corsi ITS, conferendo trasparenza e "portabilità" alle competenze acquisite dagli studenti, in particolare nell'approccio con il mercato del lavoro che, vale la pena sottolinearlo, per i percorsi di Tecnico Superiore avviene già durante la formazione, attraverso la realizzazione del tirocinio aziendale.

## 6. Risultati e prospettive

### 6.1. Il sistema di monitoraggio e valutazione degli ITS

L'avvio di un'offerta formativa in un'ottica di sistema dichiaratamente duale, non è l'unica novità introdotta nel sistema formativo post-secondario italiano con l'avvio dell'ITS. Dopo una prima fase sperimentale, l'Accordo della Conferenza Unificata del 5 agosto 2014 [14] ha avviato a regime un sistema di monitoraggio e valutazione finalizzato non solo a verificare l'efficacia delle attività messe in campo dalle Fondazioni ITS, ma a determinare:

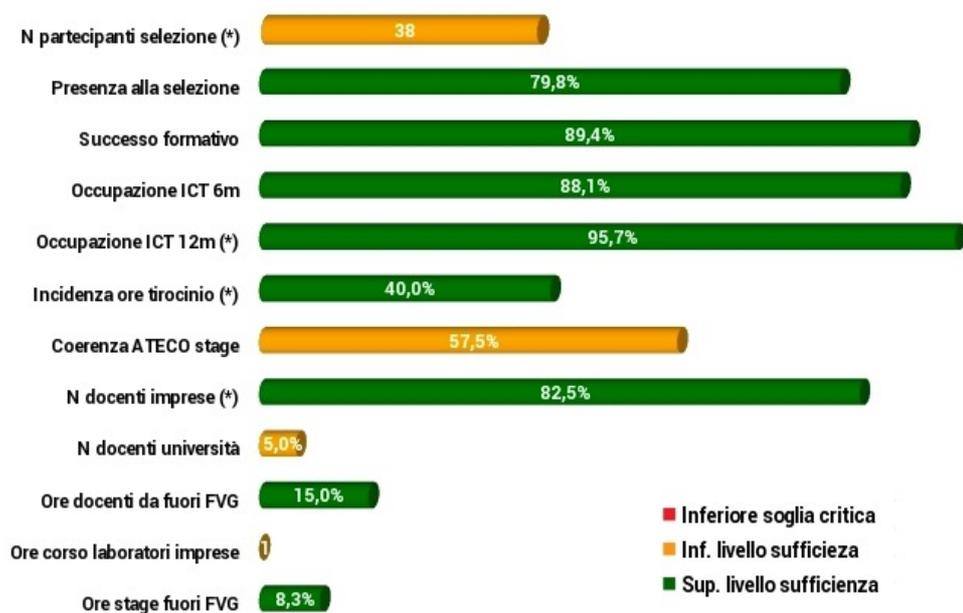
- il diritto riconosciuto alle Fondazioni ITS a rilasciare, per il tramite dell'istituto di riferimento, il titolo di Tecnico Superiore;
- l'entità dei finanziamenti di fonte MIUR, in relazione all'esito del monitoraggio stesso.

Il sistema di monitoraggio e valutazione definito dall'Accordo del 5 agosto 2014 si articola in una struttura di criteri, pesi e indici ed è stato implementato nella Banca Dati Nazionale INDIRE (Istituto Nazionale di Documentazione Innovazione e Ricerca Educativa) [15]. La Banca Dati viene alimentata da ciascuna Fondazione e produce una serie di punteggi standardizzati con soglie di criticità e sufficienza che forniscono una misura della performance di ciascun corso ITS. La Banca Dati produce inoltre un osservatorio on line [16] che fornisce dati descrittivi e statistici sui corsi e le Fondazioni ITS attivi sull'intero territorio nazionale. Condividendo pienamente tale approccio valutativo, la Fondazione ITS Kennedy ritiene che l'utilizzo degli indicatori del sistema di monitoraggio, pur migliorabile, sia il modo più trasparente di proporre i risultati ottenuti anche in contesti e finalità diversi da quelli valutativi definiti dal quadro normativo.

Il grafico della Figura 1 presenta, per i due corsi già conclusi, gli indicatori più importanti, compresi quelli per i quali è stata definita una soglia minima di criticità (evidenziati dall'asterisco). Come si può vedere, nessun indice per il nostro ITS si colloca al di sotto di tale soglia (barre in rosso, assenti), mentre due terzi di essi (barre verdi) superano la soglia della sufficienza, anche con ampio margine, come nel caso del Successo formativo, degli Esiti occupazionali e dei Docenti provenienti dal mondo delle imprese. I restanti quattro indici (barre gialle), al contrario, non raggiungono il livello della sufficienza: si tratta tuttavia di indici che, a nostro avviso, meritano una revisione. È il caso, ad esempio, di "Coerenza ATECO stage" che misura il numero di aziende stage con codice di attività ATECO coerente con il medesimo codice associato alla figura professionale di Tecnico Superiore: è difficile sostenere che uno stage svolto all'interno del dipartimento IT di un'impresa manifatturiera o del terziario non possa rappresentare una collocazione coerente per una figura professionale dell'ICT.

L'indice che riguarda il numero di partecipanti alla selezione è anch'esso meritevole di attenzione: esso evidenzia un aspetto effettivamente critico del nostro come di altri corsi ITS, ovvero la scarsa conoscenza, da parte degli studenti della scuola superiore e del sistema dell'istruzione in generale, dell'opportunità formativa rappresentata dalla formazione Tecnica Superiore,

come valida terza alternativa ai percorsi universitari e all'immediato inserimento lavorativo.



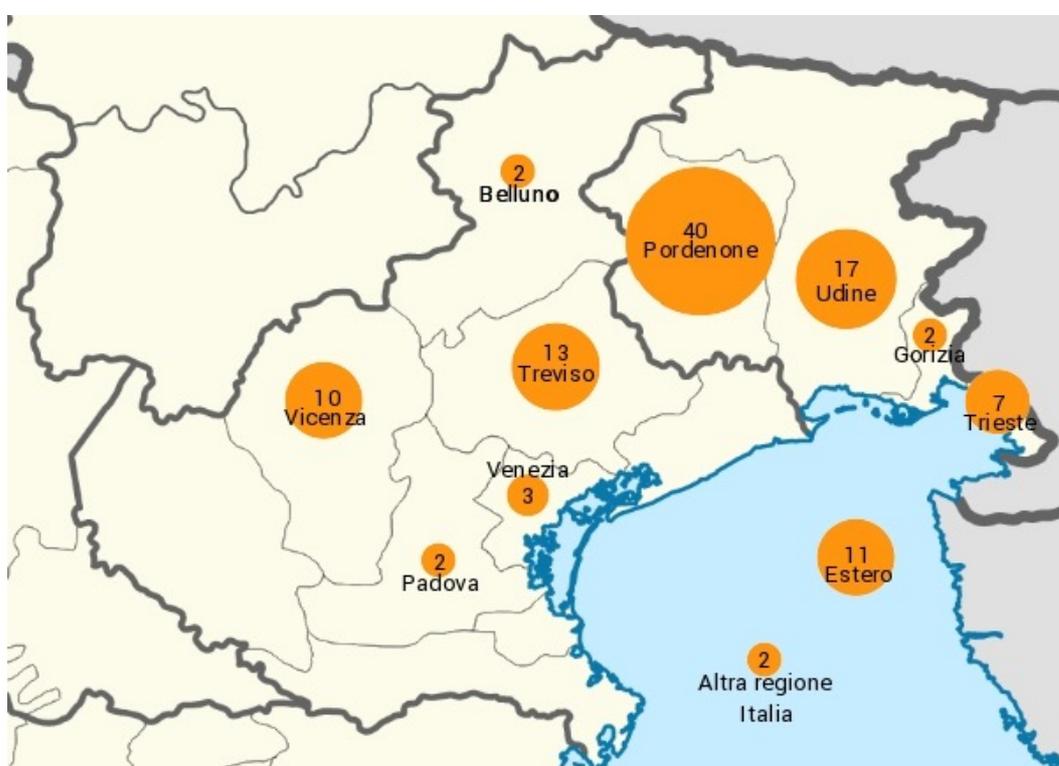
**Figura 1**  
*Indicatori del sistema di monitoraggio e valutazione, per i due corsi conclusi.*  
*Valori assoluti e %.*

In tal senso il numero ancora esiguo di Fondazioni ITS (al momento 74, sull'intero territorio nazionale e tre in Friuli Venezia Giulia), la loro nascita recentissima e la carenza della comunicazione e promozione di fonte istituzionale, costituiscono ostacoli difficilmente superabili al raggiungimento di un numero di corsisti ITS abbastanza elevato da acquisire una propria visibilità. Ad oggi infatti, l'affluenza alle selezioni di aspiranti corsisti ITS è in generale ancora modesta nonostante l'impegno di ogni singolo ITS nelle azioni di informazione, orientamento e promozione presso gli istituti superiori. Da questo punto di vista le strategie promozionali della Fondazione Kennedy si articolano in una presenza nelle scuole della regione FVG e del vicino Veneto, con seminari promozionali e presentazioni delle proprie attività, nella gestione e animazione costante di adeguati spazi sui social network, nella diffusione e nell'invio ai singoli diplomandi di depliant informativi e inviti alle selezioni.

Infine la presenza femminile tra gli aspiranti Tecnici Superiori, davvero molto bassa, conferma purtroppo che in Italia le professioni tecniche sono ancora percepite come tipicamente maschili.

## 6.2. Dimensione globale e aspetti qualitativi

Sul piano ancora quantitativo, ma non strettamente legato al sistema di monitoraggio, la dimensione globale delle attività poste in essere dalla Fondazione Kennedy, può essere descritta da alcune cifre significative: 130 studenti coinvolti nei sei corsi attivati; 42 tecnici superiori diplomati; 40 docenti utilizzati, l'85% dei quali in qualità di professionisti del settore con esperienza di lavoro in prevalenza superiore ai 10 anni ed in alcuni casi con livelli di specializzazione di livello nazionale ed esperienze di lavoro all'estero; oltre 71.000 ore di stage in azienda, un terzo delle quali in aziende residenti fuori regione e l'11% all'estero, con un coinvolgimento di 109 aziende, di cui 66 nella nostra regione, 30 in Veneto, 2 in altre regioni italiane, e 11 all'estero (Figura 2).



**Figura 2**  
Distribuzione geografica delle aziende che collaborano con la Fondazione ITS Kennedy. Valori assoluti.

Gli esiti occupazionali meritano un particolare approfondimento. Tra i 23 diplomati con il primo corso, a distanza di un anno, uno solo non risulta occupato, avendo scelto di intraprendere un percorso universitario in ICT, tutti gli altri lavorano in posizioni coerenti con il percorso di studio. Ancora in evoluzione ma promettente la situazione dei diplomati del secondo corso, per i quali l'inserimento lavorativo a sei mesi è stato positivo in 15 casi su 19. La situazione contrattuale della maggior parte degli occupati delinea un quadro di non completa stabilità, con il prevalere di contratti di apprendistato e a tempo

determinato, ma con quattro situazioni di assunzione a tempo indeterminato e ben sei casi di avvio di attività in proprio. Il valore dell'esperienza di stage, tanto dal lato dello studente che dell'impresa, è infine dimostrato da ben 15 diplomati che hanno continuato a collaborare con l'azienda di stage, con un'assunzione che, per sei ragazzi, si è concretizzata alla fine del periodo di stage, ancor prima dell'acquisizione formale del diploma.

### 6.3 Prospettive e strategie future

Per quanto concerne infine le prospettive e le strategie future della Fondazione ITS Kennedy, queste si dispiegano su quattro fronti:

- progettazione inclusiva e partecipazione proattiva agli stage e ai project work da parte delle aziende del territorio;
- consolidamento dell'opportunità di svolgere all'estero lo stage del secondo anno, della durata di 3 mesi;
- avvio di un'attività corsuale in Veneto, a fronte dell'autorizzazione all'erogazione di corsi ITS ottenuto lo scorso anno attraverso la partnership con l'ITT Chilesotti di Thiene (VI);
- avvio, con l'anno formativo 2016-2017 di un'offerta formativa articolata su sei semestri, ovvero tre anni scolastici.

L'emergere di alcune criticità nel processo che va dall'individuazione delle figure professionali alla stesura del progetto vero e proprio del corso, parallelamente alla necessità di anticipare il più possibile la fase promozionale dei corsi, al fine di superare la criticità riscontrata nella raccolta delle iscrizioni, fa sì che il "time-to-market" dal progetto al collocamento dei diplomati sul mercato del lavoro si avvicini ai tre anni. Un tempo non compatibile con la scelta della Fondazione di muoversi secondo lo stato dell'arte delle tecnologie ICT. Ciò ha comportato una ridefinizione della strategia progettuale, tuttora in fase di implementazione:

- con un coinvolgimento delle aziende del territorio nella fase di individuazione dei fabbisogni e dei trend di sviluppo del settore con la metodologia dei focus group;
- con un'inversione concettuale di ruoli e posizioni nell'organizzazione degli stage, invitando le aziende, in un ottica di call-to-proposal, a proporre micropoggetti di innovazione/sperimentazione da realizzare in sede di stage e con project works;
- con il riferimento esplicito e diretto alle e-CF in sede di individuazione delle figure professionali e delle declinazioni tecnologiche e in sede di definizione degli obiettivi di stage e di project work.

Durante il corrente anno formativo 17 studenti del secondo anno, grazie all'approvazione di un progetto Erasmus+ KA1, hanno svolto lo stage in aziende di cinque paesi dell'Unione Europea (Portogallo, Regno Unito, Slovenia, Spagna, Svezia). Per il prossimo anno la Fondazione ha presentato un nuovo progetto per il coinvolgimento di 28 ragazzi con l'allargamento ad altri due paesi (Francia e Germania) ed il diretto coinvolgimento in partnership di due aziende

multinazionali. Più in generale la Fondazione intende estendere il proprio bagaglio di esperienze e il network di relazioni sul piano internazionale, partecipando:

- in qualità di Ente Intermedio ad un progetto KA1, per collocare in stage nella propria rete di aziende alcuni studenti portoghesi;
- ad un bando di candidatura, di imminente scadenza, per l'acquisizione dell'Erasmus VET Charter 2015, per consolidare in maniera stabile l'offerta ai propri studenti di esperienze di mobilità all'estero;
- come passo successivo, alla candidatura per un progetto Erasmus KA2-Partenariati Strategici 2016, in partnership con imprese e istituzioni per la sperimentazione e l'approfondimento sul piano internazionale di metodologie e tools per la didattica dell'ICT e, in particolare, per l'applicazione del framework europeo e-CF in ambito formativo.

L'estensione dei corsi alla terza annualità, infine, si configura, sul piano istituzionale e normativo, come la concretizzazione di quanto previsto negli atti legislativi che hanno sancito la nascita dei percorsi ITS: già citati D.P.C.M. 25 gennaio 2008 e D.M. MIUR 7 settembre 2011. L'articolo 4 comma 5 di quest'ultimo, infatti, dichiarava che, nel caso "di percorsi della durata di sei semestri, le competenze finali sono riferibili a un livello del Quadro europeo delle qualifiche per l'apprendimento permanente (EQF) superiore al livello al quale sono riferibili le competenze finali relative ai percorsi della durata di quattro semestri" (ovvero quinto livello EQF). Successivamente però, né a livello nazionale né regionale, alcun provvedimento ha reso esplicita la possibilità di avviare percorsi di sei semestri, disponendo l'indispensabile supporto finanziario. Solo il recentissimo documento "La buona scuola", nella sua parte quinta, nel dare forte rilievo al segmento terziario del sistema duale (ITS), sembra riprendere questa previsione normativa che permette di completare la formazione tecnica superiore con un titolo di studio equivalente nella scala EQF alla laurea triennale.

Sul piano concreto delle strategie di marketing e degli obiettivi didattici e di competenza, l'offerta di corsi di sei semestri, equipollenti ai percorsi universitari brevi, costituirebbe un salto qualitativo, forse in grado di sdoganare la formazione tecnica da un ruolo subalterno ai percorsi accademici, fornendole la dignità che la prassi di molte start up digitali, le analisi sulla cronica carenza di professionalità tecniche in ICT [17][18] e la propulsività economica dell'innovazione digitale le hanno già implicitamente assegnato.

## Bibliografia

[1] <http://hubmiur.pubblica.istruzione.it/web/istruzione/dg-ifts/area-iefp> (ultimo accesso aprile 2015)

[2] D.P.C.M. 25 gennaio 2008 "Linee Guida per la riorganizzazione del sistema di Istruzione e Formazione Tecnica Superiore e la costituzione degli Istituti tecnici superiori"

- [3] [http://ec.europa.eu/ploteus/sites/eac-eqf/files/leaflet\\_en.pdf](http://ec.europa.eu/ploteus/sites/eac-eqf/files/leaflet_en.pdf) (ultimo accesso aprile 2015)
- [4] Raccomandazione 2008/C 111/01/CE del Parlamento Europeo e del Consiglio del 23 aprile 2008
- [5] Presidenza del Consiglio dei Ministri, Conferenza permanente per i rapporti tra lo Stato, le Regioni e le Province Autonome di Trento e Bolzano, "Accordo sulla referenziazione del sistema italiano delle qualificazioni al quadro europeo delle qualifiche per l'apprendimento permanente (EQF)"
- [6] D.M. Ministero dell'Istruzione dell'Università e della Ricerca 7 settembre 2011 recante "norme generali concernenti i diplomi degli Istituti Tecnici Superiori (I.T.S.) e relative figure nazionali di riferimento, la verifica e la certificazione delle competenze"
- [7] <http://www.eucip.it> (ultimo accesso aprile 2015)
- [8] <http://www.aicanet.it/aica/ePMQ/per-i-candidati> (ultimo accesso aprile 2015)
- [9] <http://www.eqdl.it> (ultimo accesso aprile 2015)
- [10] <http://www.ecompetences.eu/> (ultimo accesso aprile 2015)
- [11] <http://profiletool.ecompetences.eu/> (ultimo accesso aprile 2015)
- [12] D.Lgs. 16 gennaio 2013, n 13, "Definizione delle norme generali e dei livelli essenziali delle prestazioni per l'individuazione e validazione degli apprendimenti non formali e informali e degli standard minimi di servizio del sistema nazionale di certificazione delle competenze"
- [13] IX Commissione - Istruzione, Formazione e Lavoro – della Conferenza delle Regioni, 10 luglio 2013, "Piano di lavoro per l'implementazione del Repertorio nazionale dei Titoli di istruzione e formazione e delle Qualificazioni professionali - art. 8 del decreto legislativo n. 13/2013 - verso l'attuazione del Sistema nazionale di certificazione delle competenze"
- [14] Conferenza Unificata della Presidenza del Consiglio dei Ministri, 5 agosto 2014, "Accordo tra Governo, Regioni ed Enti Locali, per la realizzazione del sistema di monitoraggio e valutazione dei percorsi degli Istituti Tecnici Superiori"
- [15] [http://www.indire.it/its/content/index.php?action=lettura2Col&id\\_m=11730&id\\_cnt=14958](http://www.indire.it/its/content/index.php?action=lettura2Col&id_m=11730&id_cnt=14958) (ultimo accesso aprile 2015)
- [16] <http://www.indire.it/its/> (ultimo accesso aprile 2015)
- [17] Renzo Provedel (2002), "Carenza di competenze: "Skill Shortage" nel settore ICT", Mondo Digitale, n.1, marzo 2002
- [18] DG Employment, Social Affairs & Inclusion of the European Commission, European Vacancy Monitor, February 2014, n. 12, pag. 7 <http://ec.europa.eu/social/BlobServlet?docId=11426&langId=en> (ultimo accesso aprile 2015)
- [19] Assintel (2014), "Assintel report 2014 - Il mercato del software e servizi in Italia", <http://www.assintel.it/wp-content/uploads/2014/10/Assintel->

Report-2014.pdf, [https://prezi.com/yfyvdlmrxsr4/milano\\_assintel-report-2014/](https://prezi.com/yfyvdlmrxsr4/milano_assintel-report-2014/)  
(ultimo accesso aprile 2015)

[20] Assinform (2014), "Rapporto Assinform sull'informatica le telecomunicazioni e i contenuti multimediali", <http://www.areeriservate.assinform.it/ImagePub.aspx?id=159424>, <http://www.areeriservate.assinform.it/ImagePub.aspx?id=166173>  
(ultimo accesso aprile 2015)

## Biografia

**Valerio Libralato**, laureato in sociologia a Trento nel 1986. Si è occupato per quindic'anni di indagini empiriche su questionario in campo socio-economico. Dal 2001 lavora presso IALFVG in cui ha realizzato analisi di settore, valutazione di progetti di formazione e ricerca e progetti formativi, tipicamente per la formazione post-secondaria nel settore ICT.

Dalla suo avvio è impegnato con la Fondazione ITS Kennedy come responsabile della progettazione e, dal 2012, come membro del CTS, contribuendo in particolare all'adozione dello standard e-CF.

email: [valerio.libralato@tecnicosuperiorekennedy.it](mailto:valerio.libralato@tecnicosuperiorekennedy.it)