

# Intervista a Frank Mittelbach

*Frank Mittelbach, Gianluca Pignalberi, Dave Walden*

## Sommario

Sia a Free Software Magazine (FSM, <http://www.freesoftwaremagazine.com>) che al T<sub>E</sub>X Users Group (TUG, <http://www.tug.org>) piace pubblicare interviste. Recentemente, Gianluca Pignalberi di FSM e Dave Walden del TUG hanno entrambi contattato Frank Mittelbach per intervistarlo. Aniché fare due interviste separate, Mittelbach, Pignalberi e Walden si sono accordati per un'intervista congiunta, in modo da soddisfare gli interessi comuni già condivisi da FSM e TUG.

## Abstract

Free Software Magazine (FSM, <http://www.freesoftwaremagazine.com>) and the T<sub>E</sub>X Users Group (TUG (<http://www.tug.org/>)) both like to publish interviews. Recently, Gianluca Pignalberi of FSM and Dave Walden of TUG both approached Frank Mittelbach about interviewing him. Rather than doing two separate interviews, Mittelbach, Pignalberi, and Walden decided on a combined interview in keeping with the mutual interests already shared by FSM and TUG.

*Dave Walden, cointervistatore: Frank, per piacere, inizia parlandoci un po' di te e di come sei entrato in contatto con L<sup>A</sup>T<sub>E</sub>X.*

Frank Mittelbach, intervistato: Vivo con la mia famiglia a Magonza (Germania) dai primi anni Ottanta, cioè per la maggior parte della mia vita, finora. Oltre al mio hobby principale (l'arte tipografica), che può veramente essere definito il mio secondo lavoro, mi piace giocare a bei giochi da tavola, ascoltare musica jazz e leggere (principalmente letteratura inglese). Professionalmente lavoro per la Electronic Data Systems, dove attualmente sono responsabile della concezione e dell'implementazione di controlli remoti e gestione di reti e sistemi distribuiti.

Quando ero studente di matematica e informatica all'Università Gutenberg di Magonza sono entrato in contatto prima con T<sub>E</sub>X e poi con L<sup>A</sup>T<sub>E</sub>X; questo ha contribuito al mio interesse per la composizione tipografica ed in particolare alla ricerca sugli algoritmi per la composizione automatica di alta qualità.

Durante i miei giorni da studente, negli anni Ottanta, un amico riportò dall'università di Stanford un nastro di sorgenti contenente qualcosa come T<sub>E</sub>X 1.1 e notizie affascinanti sulla qualità del programma (fino ad allora scrivevamo le nostre tesi battendole a macchina, incollando a mano i simbo-

li o, come nel caso di alcune sofisticate macchine da scrivere IBM, cambiando la "palla" ogni due secondi). Provò ad implementare il programma sul sistema Multics che avevamo all'università e ci riuscì: quella era probabilmente la prima, se non l'unica, implementazione di T<sub>E</sub>X per questo sistema.

In questo modo fui introdotto a T<sub>E</sub>X e  $\mathcal{A}\mathcal{M}\mathcal{S}$ -T<sub>E</sub>X e scrissi il mio primo articolo, ottenendo uno splendido risultato. L'unico inconveniente fu che

$$\sum_{i=1}^n x_i$$

il nastro di Stanford conteneva solo font Quasi Computer Modern alla risoluzione di 200 dpi (e senza METAFONT), e l'unica stampante grafica che avevamo a disposizione aveva una risoluzione di 72 dpi. Quindi l'output che ottenevamo era grande come la formula al centro di questo capoverso (o più grande): stupendo per tappezzare le pareti, ma non proprio adatto ad una tesi da consegnare. Come risultato, il mio amico ha dovuto scrivere la sua tesi nel modo tradizionale, nonostante i suoi sforzi.

Qualche tempo dopo mi fu chiesto dal mio dipartimento di installare una versione commerciale di T<sub>E</sub>X sui nostri PC nuovi fiammanti e di tenere una serie di lezioni a studenti e professori su come usarlo. Questa distribuzione era provvista di L<sup>A</sup>T<sub>E</sub>X 2.08 e una copia con rilegatura mobile del manuale (che sarebbe diventato il libro di Leslie Lamport su L<sup>A</sup>T<sub>E</sub>X). L<sup>A</sup>T<sub>E</sub>X, confrontato a Plain T<sub>E</sub>X, mi sembrava molto buono ma, purtroppo,



FIGURA 1: Frank Mittelbach

se usato per produrre un documento, si piantava durante il caricamento dello “stile di documento articolo” a causa di mancanza di memoria su quei PC. Così il mio approccio a LATEX si fermò dopo la lettura del manuale, e fui costretto a sviluppare un mio pacchetto di macro che implementava concetti analoghi, richiedendo al contempo meno memoria. Un anno dopo fu possibile usare realmente LATEX nel dipartimento, ed ho ritirato il mio pacchetto.

Questo esercizio iniziale mi ha dato una buona visione dei concetti e dei meccanismi interni di un sistema come LATEX e mi ha in seguito permesso di criticare costruttivamente certi aspetti di LATEX – cosa che forse ha spinto Leslie a passarmi lo sviluppo e il mantenimento di LATEX.

*Gianluca Pignalberi, cointervistatore: Molti dei nostri lettori hanno familiarità con LATEX ma, per quelli che non lo fossero, puoi presentarglielo?*

FM: LATEX è un sistema di composizione non interattivo che usa TEX o una delle sue varianti ( $\epsilon$ -TEX, PDFTEX, Omega) come motore di composizione.

Il programma TEX (sviluppato dal prof. Donald Knuth all’inizio degli anni Ottanta) è un motore di composizione a basso livello programmabile, i cui concetti e algoritmi forniscono una conoscenza di microtipografia<sup>1</sup> della più alta qualità in tempi, come quelli attuali, in cui questa conoscenza è in lento declino per via del fatto che sempre più autori sono costretti a diventare grafici e tipografi di se stessi, pur senza un’adeguata formazione. TEX è in special modo famoso per il suo eccellente algoritmo di interruzione di riga [correlato alla sillabazione  $NdT$ ] e per le sue capacità di composizione di formule matematiche, entrambi insuperati, nonostante la libera disponibilità del programma e dei suoi algoritmi da oltre vent’anni.

LATEX è un pacchetto di macro, scritto per il motore TEX, che permette all’utente di astrarsi dalle capacità di formattazione a basso livello di TEX fornendo interfacce di più alto livello, le quali permettono all’autore di etichettare il testo con etichette logiche invece che procedurali (ad esempio, specificando che qualcosa è una lista o un paragrafo, piuttosto che affermare che qualcosa dovrebbe essere composto con un carattere nero e dello spazio sopra e sotto). La vera trasformazione di un sorgente LATEX in un documento finito viene fatta con l’aiuto di “fogli di stile” e accomodamenti di configurazione che permettono cambiamenti anche radicali dell’aspetto in maniera consistente, e senza toccare o cambiare il sorgente<sup>2</sup>.

1. La microtipografia riguarda l’aspetto dettagliato di caratteri e spaziature, ad esempio la crenatura (allargamento o restrizione dello spazio) tra le lettere, la generazione e il posizionamento delle legature, le interruzioni di riga, ecc. Per contro, la macrotipografia riguarda strutture più grandi, come il disegno di testate, liste o pagine.

2. Be’, idealmente, ma si veda oltre.

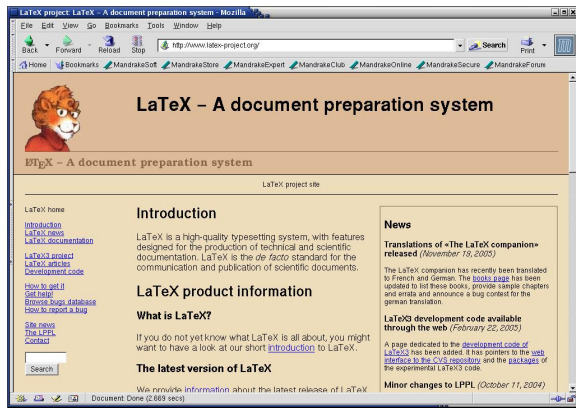
Storicamente, LATEX è stato largamente influenzato da un sistema chiamato Scribe (di Brian Reid). A sua volta, il concetto di etichetta logica di LATEX è stato abbastanza influente per i DTD dell’HTML e dei vari SGML/XML, così come lo furono i suoi approcci per convertire queste etichette logiche in una rappresentazione visuale.

Una delle differenze tra LATEX e molti altri approcci simili è che il linguaggio LATEX è effettivamente a sviluppo comunitario: nuovi pacchetti che aumentano (o modificano) le etichette e le funzioni di composizione appaiono continuamente, così che ormai LATEX offre soluzioni di composizione per quasi ogni argomento, pure tanto diversi quanto composizione di giochi (come scacchi, *go* o parole incrociate), formule chimiche o musica. Un’altra differenza importante è che, sebbene LATEX abbia divulgato il concetto di etichetta logica, fornisce anche dei modi per regolare finemente i risultati (essenzialmente fornendo interfacce verso l’etichettatura procedurale), accettando il fatto che nessuna trasformazione automatica di etichette logiche in rappresentazioni visuali può risolvere automaticamente tutti i problemi generati da restrizioni fisiche del formato di output (come la larghezza di una riga di testo o la dimensione della pagina). Mentre in alcune applicazioni questa regolazione fine non aggiunge valore (come nella pubblicazione di contenuti di database, dove è richiesta la totale automazione), è cruciale per comporre libri e articoli di riviste di alta qualità.

*GP: Quante persone fanno parte ufficialmente del LATEX Project? E... come definiresti il “LATEX Project”?*

FM: Il team del LATEX Project è un gruppo di persone abbastanza ristretto (in lento cambiamento), che badano al kernel di LATEX e ad un piccolo numero di pacchetti fondamentali che costituiscono una solida base per una grossa quantità di pacchetti e aggiunte in costante evoluzione. Fornire e sorvegliare un nucleo stabile è un compito importante (sebbene non condiviso da molti) per mantenere LATEX vivo come linguaggio di interscambio di documenti. Tra i membri che hanno composto, o attualmente compongono, il team ricordiamo Javier Bezos, Johannes Braams, David Carlisle, Michael Downes, Denys Duchier, Robin Fairbairns, Morten Høgholm, Alan Jeffrey, Thomas Lotze, Chris Rowley, Rainer Schöpf, and Martin Schröder con vari gradi di coinvolgimento.

Storicamente il progetto è subentrato nel 1991 a Leslie Lamport per la manutenzione e lo sviluppo di LATEX 2.09. Fino ad allora il sistema era stato diviso in diverse varianti incompatibili, che spesso proibivano la corretta elaborazione da parte di una variante di LATEX di documenti creati con un altro LATEX, sebbene l’indipendenza dai sistemi fosse originariamente uno dei risultati importanti di LATEX come linguaggio di documentazione per la

FIGURA 2: La home page del LATEX Project

comunità scientifica. Un altro risultato per il team fu quello di correggere apparenti deficienze nei concetti di LATEX 2.09. Il team ha trattato entrambe le questioni all’inizio degli anni Novanta con LATEX 2<sub>ε</sub>, che ha fornito una piattaforma stabile e consolidata in grado di fornire ulteriori possibilità di sviluppo al di fuori del codice del kernel.

Sebbene LATEX 2<sub>ε</sub> curasse la maggior parte, se non tutte, le deficienze trovate nel primo decennio di uso di LATEX 2.09, si è originariamente pensato che LATEX 2<sub>ε</sub> sarebbe stato solo un passo intermedio verso LATEX 3. Ma, nel tempo, è diventato sempre più chiaro che le rimanenti questioni aperte non sarebbero potute essere adeguatamente risolte entro i limiti imposti da: a) TEX come motore di composizione, e b) nessuna modifica ai concetti fondamentali realizzati dentro LATEX.

Come risultato, la maggior parte degli sforzi fatti negli anni recenti dai membri del team di LATEX è stata volta a ricercare le caratteristiche desiderabili per i motori di formattazione sottostanti e a sviluppare linguaggi e concetti sperimentali per un’interfaccia alla composizione tipografica per i grafici, livello di astrazione largamente mancante dall’attuale LATEX (che attualmente richiede spesso la programmazione in TEX).

Dunque, attualmente, una definizione del LATEX Project potrebbe essere che questo lavora per fornire le fondamenta dei concetti e l’implementazione di base di un nuovo sistema di composizione basato sugli aspetti positivi di LATEX 2<sub>ε</sub> (ad esempio, etichette logiche, estensibilità) ma, a livello di codice, non necessariamente orientato a fornire compatibilità. A livello di sintassi dei documenti la situazione è, ovviamente, differente, poiché il riuso dei documenti più vecchi è senz’altro un obiettivo importante. Ma, anche là, la principale concentrazione sarà sulla chiarezza dei concetti; come risultato, la compatibilità potrebbe essere ristretta in determinati casi a fornire un supporto alla conversione automatica.

*DW: Nel mondo di TEX hai lunghi precedenti di lavoro collaborativo, in special modo con Rainer*

*Schöpf, agli albori di LATEX. Deve proprio piacerti il lavoro collaborativo. Raccontaci qualcosa sui tuoi approcci a questo metodo.*

FM: In effetti mi piace collaborare; nel corso degli anni ho lavorato con successo con molte persone diverse (su vari argomenti e in differenti campi). Per me, il valore principale della collaborazione sta nella fase di sviluppo delle idee che, a mio parere, sono meglio elaborate nell’ambito di uno scambio aperto. La mia immagine mentale, in questo caso, è un gioco tipo il ping pong, che si sviluppa solo se si reagisce direttamente a ciò che pensa l’avversario e si prende la palla man mano che viene giocata. La gente che ha lavorato con me sa che mi piacciono le sessioni disegnate alla lavagna bianca (ho bisogno di visualizzare lo svolgersi del gioco) e i metodi di *brainstorming* e delle mappe mentali.

Ma sono anche fissato con i dettagli, e posso impiegare molte energie e sforzi per finire veramente qualcosa (per soddisfazione personale), quando lo considero meritevole. La collaborazione a quel livello – dopo che il lavoro di sviluppo di concetto e progetto è terminato e il molto lavoro sul dettaglio comincia – assume normalmente una delle due possibili forme: o mi relego il più possibile al ruolo di mentore e lascio lavorare gli altri alle implementazioni vere e proprie, o metto così tante energie in un determinato compito che queste superano di un fattore grande il coinvolgimento degli altri. Il mio motto è “Es gibt nichts Gutes, außer man tut es” (traduzione libera: Non verrà fatto niente di buono finché non lo fai di “buono buono”) di Erich Kästner che, almeno in tedesco [e in italiano *NdT*], fa rima.

Un sacco di collaborazione avviene necessariamente per email (a causa dei diversi paesi in cui si vive, ecc.), ma trovo estremamente prezioso interrompere questo metodo di lavoro, a intervalli irregolari, con incontri *de visu* per tirare fuori le idee e renderle concrete quel tanto che basta per andare avanti in semi-isolamento, con le sole email e/o telefonate come mezzo di comunicazione “diretta”. Ciò spiega anche perché la gran parte del lavoro fondamentale, associato a Rainer Schöpf e me, risale al tempo in cui eravamo entrambi studenti universitari e avevamo modo di avere regolarmente scambi di idee davanti ad una lavagna bianca (bevendo galloni di tè).

In generale ritengo che Frederick Brooks ha ragione quando nel libro *The Mythical Man-Month* BROOKS, JR (1995) sostiene che, per avviare un progetto software di successo, hai bisogno di una squadra di persone relativamente piccola e strutturata, che sia responsabile di decidere il progetto finale. Un grande “comitato” di progetto porta solo a cattivi risultati, per arrivare a troppi compromessi tra fazioni differenti, o per l’incorporazione di concetti progettuali incompatibili.

GP:  $T_{E}X$  è diffusamente considerato il miglior sistema di composizione tipografica, ma i compositori professionisti sembrano preferire l'uso di software visuale commerciale. Perché consiglieresti loro di usare  $L^A T_{E}X$  invece di un altro sistema? O, al contrario, perché no?

FM: Non c'è dubbio che  $T_{E}X$  abbia qualità superiori in un buon numero di campi rispetto ad altri sistemi di composizione (ad esempio, la qualità della divisione dei capoversi, o la presentazione di formule matematiche, ecc.), ma è stato progettato come un programma di elaborazione batch e ciò implica il non permettere l'interazione con l'utente. Nei sistemi di composizione WYSIWYG l'utente può fare correzioni visuali, che sono dunque mostrate istantaneamente, mentre con  $T_{E}X$  (o  $L^A T_{E}X$ ) devi modificare il tuo sorgente e poi riprocessarlo e controllare che le tue correzioni abbiano prodotto l'effetto desiderato.

Specialmente nell'ultima fase della produzione di un libro, la tendenza di  $T_{E}X$  a raggiungere l'alta qualità solo per mezzo di cambiamenti non immediatamente chiari può veramente essere un difetto anziché un pregio (almeno se non consideri il perché e adatti il tuo metodo di lavoro). Per esempio, a causa dell'ottimizzazione globale della suddivisione dei capoversi, la *rimozione* di una singola parola in un capoverso determinerà un completo riallestimento dell'intero capoverso. Ciò potrebbe, in realtà, rendere il capoverso addirittura una riga *più lungo*, se  $T_{E}X$  decidesse che un posizionamento più lasco di tutte le righe produrrebbe la migliore soluzione possibile. Se ciò accadesse durante la correzione dei refusi all'ultimo minuto, in un documento altrimenti ben formattato, sarebbe una bella seccatura.

La maggior parte dei grafici e dei compositori professionisti sono abituati a lavorare visivamente con risposta e controllo immediati. Dunque, ad essi, sistemi come  $L^A T_{E}X$  sembrano difficili da maneggiare e non vedono benefici in questi modelli di lavoro estranei.

Se l'uso di  $L^A T_{E}X$  sia consigliabile o meno dipende fortemente dal lavoro che si ha sotto mano, e la risposta è sempre dipendente dal contesto. In breve, suggerirei di usare  $L^A T_{E}X$  o sistemi simili quando uno o più fattori tra i seguenti giochi un ruolo importante nell'ambito del lavoro:

- preferenza dell'utente a pensare per strutture logiche;
- allestimenti che richiedono consistenza;
- documenti che non hanno un allestimento interamente definito o saranno presentati parallelamente in diverse vesti;
- documenti che richiedono suddivisione in capoversi di alta qualità;
- documenti che contengono molta matematica;

- contenuto prodotto automaticamente (ad esempio, proveniente da database);
- materiale corposo.

D'altro canto, i seguenti fattori fanno pendere l'ago della bilancia verso l'uso di un sistema visuale (di buona qualità!):

- preferenza dell'utente a pensare per strutture visive;
- utente non del tutto a suo agio a lavorare con linguaggi di programmazione (un front-end di alto livello per  $T_{E}X$ , ad esempio  $L^A T_{E}X$  o  $ConT_{E}Xt$ , aiuta, ma...);
- allestimenti che richiedono più flessibilità visuale che consistenza (ad esempio, le intestazioni sono disegnate una per una in base agli oggetti vicini);
- allestimenti che richiedono che il testo scorra intorno a forme arbitrarie ( $T_{E}X$  non è stato progettato per questo);
- allestimenti che cambiano la misura orizzontale da colonna a colonna;
- materiale non corposo.

Quello che realmente fa la differenza può variare in base alle circostanze. Per esempio, in *The L<sup>A</sup>T<sub>E</sub>X Companion* MITTELBACH e GOOSSENS (2004), con i suoi circa 1000 esempi nel testo, è stato un enorme vantaggio poter riallestire l'aspetto degli esempi senza modificarli individualmente.

DW: C'è qualche relazione tra TUG, DANTE, o qualsiasi altro  $T_{E}X$  Users Group e il  $L^A T_{E}X$  Project, e come i vari gruppi di utenti potrebbero mantenere la vitalità di  $T_{E}X$  e dei suoi derivati?

FM: La relazione tra il  $L^A T_{E}X$  Project e i  $T_{E}X$  Users Group può probabilmente essere ben descritta come lasca e informale. Diverse attività del team di progetto sono state supportate, in un modo o nell'altro, da un gruppo di utenti (ad esempio, fornendo spazi di incontro ad un convegno). Comunque, la maggior parte delle attività del team ha contato su risorse diverse, come il supporto dello ZDV (il laboratorio di calcolo dell'Università Gutenberg di Magonza), pagamenti dei diritti di *The L<sup>A</sup>T<sub>E</sub>X Companion*, e in piccola parte contributi dei singoli utenti. La maggiore alleanza con un gruppo di utenti è stata probabilmente lo sviluppo di un modello per un insieme di caratteri matematici, per cui il gruppo di utenti francese ha finanziato uno studente per tre mesi per lavorare con me su questo argomento. Il risultato di questo lavoro ZIEGLER (1994) sta finalmente portando alcuni frutti, dato che è stato d'aiuto per lo sviluppo del font STIX.

Comunque, i gruppi di utenti sono estremamente importanti per progetti come il LATEX Project, in quanto forniscono una struttura di ricerca per contatti e discussioni faccia a faccia in convegni e pubblicazioni su rivista. Non può essere sottovalutato questo aspetto di fornire un ambiente di ricerca. E, se il gruppo di utenti non fosse più in grado di fornirla, si potrebbe arrivare anche alla morte della comunità. Certamente mi piace (e sono sicuro piaccia anche ad altri) lo scambio fruttuoso che è possibile solo in questa struttura.

È difficile rispondere alla domanda su che cosa i gruppi di utenti possano fare a lungo termine per sostenere la vitalità di TEX e i suoi derivati – sfortunatamente, non sono sicuro che gli stessi gruppi di utenti sopravvivano a lungo termine. Il ruolo delle organizzazioni di utenti è cambiato negli ultimi venti anni. All’inizio, solo ottenere un sistema TEX installato era lo sforzo principale, e i gruppi di utenti erano formati da persone interessate ad aiutarsi reciprocamente, a scambiarsi conoscenze e idee, e a supportare lo sforzo di sviluppo. Allora il ruolo dei gruppi di utenti era piuttosto chiaro e il beneficio per ogni membro era immediatamente visibile, ad esempio, ottenere informazioni altrimenti non disponibili, avere aiuto, ecc.

Col passare del tempo, l’accesso alle installazioni TEX e ai suoi accessori (come pacchetti LATEX, ecc.) è drasticamente cambiato – le moderne installazioni sono preimpacchettate, il software è disponibile in grandi archivi, e c’è molta documentazione disponibile. Come risultato, il tipico utente TEX/LATEX non ha bisogno di capire i meccanismi sottostanti e (sfortunatamente) non è più interessato a partecipare al loro sviluppo – gli utenti sono cambiati molto: da membri di un gruppo di persone con interessi simili che partecipano attivamente, eccitate dalla possibilità di fare composizione tipografica di alta qualità, a utenti finali di un “prodotto finito” che diventano molto tristi se il prodotto non fa esattamente ciò che gli utenti vogliono che faccia. Per questo nuovo tipo di utenti i gruppi di utenti non giocano un ruolo importante poiché, in questo momento, i gruppi di utenti non hanno risorse per aiutare fattivamente le persone nei loro problemi.

Ciò è in qualche modo ironico, dato che sono stati perlopiù i membri dei gruppi di utenti ad iniziare tutti quei cambiamenti che ora sembrano condurre alla fine della missione dei gruppi di utenti e della ragione di esistere agli occhi dell’utente tipo.

In teoria, penso che il modo migliore in cui i gruppi di utenti possano aiutare attualmente sia lavorare nelle seguenti aree:

- assumere e fornire le risorse che mantengano il “prodotto” vivo e ben mantenuto;
- fornire un forum adatto per la comunità di sviluppo;
- ottenere e gestire fondi di ricerca;

- attrarre nuovi utenti per allargare la base.

Sfortunatamente, nessuno di questi obiettivi è facilmente raggiungibile. Ci sarebbe bisogno di capitali (e risorse) consistenti, oltre quelli già a disposizione dei gruppi, e non è chiaro come ciò, come un charter, potrebbe attrarre abbastanza membri nuovi, che potrebbero dunque dividersi quei costi. Dopotutto, alla maggior parte delle persone il “prodotto” e il suo supporto sembra essere disponibile gratis, così, il pensiero degli utenti attuali è: “Perché pagare una (sostanziale) quota associativa periodica, quando tutto quello di cui si ha bisogno è di collegarsi a Internet e porre una domanda su `comp.text.tex` o scaricare qualche software dal CTAN? Io non vado a quei convegni a cui ‘loro’ partecipano, dunque, perché dovrei finanziarli?” Sarebbe necessario contrastare questo pensiero e far capire alla gente perché i gruppi di utenti sono, nonostante tutto, benefici per loro; ma, sfortunatamente, molte persone approfittano di un pasto gratis se possono, senza considerare le conseguenze.

*GP: Il tuo progetto ha generato una licenza: la LATEX Project Public License. Grazie alle ultime modifiche ad essa, LATEX può veramente essere considerato un vero software libero. Come questo fatto ha incrementato la diffusione di questo strumento?*

FM: Ad essere onesti, ho sempre considerato e tuttora considero LATEX *vero software libero*, indipendentemente dalla licenza sotto cui era distribuito in passato o è distribuito ora. Il termine “libero” ha significati decisamente diversi per persone diverse, e non sono necessariamente d’accordo con l’idea di alcune persone che la loro libertà di poter cambiare arbitrariamente le cose senza restrizioni dovrebbe essere considerata un bene più importante del diritto di altri ad ottenere quello che si aspettano quando usano un certo prodotto.

LATEX non è semplicemente un prodotto per un singolo utente, ma un linguaggio usato per la comunicazione di informazioni e uno dei punti più importanti è che permette di elaborare un documento in posti diversi con risultati identici, a patto che sia usata la stessa versione di LATEX. Questa è una caratteristica su cui la gran parte della comunità fa affidamento, così la LPPL (LATEX Project Public License) originale LATEX3 PROJECT TEAM (2003) diceva, in breve: fate qualunque cosa vogliate del file X ma, se lo cambiate, (per esempio modificate il suo comportamento nel sistema) cambiategli nome in qualcos’altro, cosicché le persone che si affidano alla caratteristica della comunicazione di LATEX non siano affette dalle vostre modifiche.

Tecnicamente, questo permetteva qualunque modifica e cambiamento desiderati, ma dava agli utenti di LATEX la scelta informata se usare o meno una versione modificata con i propri documenti. In alcuni casi, ciò avrebbe comportato alcuni sforzi



Ovviamente, hai un percorso record nel realizzare progetti TEXosi grandi e complessi. Intravedi di venire coinvolto in qualcosa come il progetto exTEX o nell'inizio del tuo progetto di implementazione a basso livello per un miglioramento di TEX?

FM: Quando ho scritto *E-TEX: Guidelines for future TEX extensions* MITTELBACH (1990) nel 1990, il tempo non era ancora maturo per migliorare TEX e molte persone hanno considerato veramente un affronto a Don il fatto che avessi suggerito che ci potesse essere qualcosa da migliorare nel suo prodotto (ricordo, per esempio, pubbliche fantasterie circa strane teorie di un oscuro e sconosciuto tipografo tedesco. Be', quelle "strane teorie" mi sono state suggerite da nessun altro che Hermann Zapf che, sebbene tedesco, non può essere precisamente chiamato sconosciuto, meno che mai oscuro). Comunque, sia come sia, quello che stavo sfidando in quell'articolo era la qualità di composizione a livello di microtipografia. In ogni caso, dato che TEX era tecnicamente molto meglio di qualunque altra cosa all'epoca, la mia sfida fu probabilmente prematura, e ci è voluto circa un decennio prima che fossero condotti i primi veri esperimenti a quel livello e le cose avanzassero in quel campo (per gran parte con lo sviluppo di versioni sperimentali di PDFTEX, ma anche codice sperimentale di altri, come Matthias Clasen).

Discutendo dei miglioramenti a TEX bisogna distinguere tre aree fortemente disgiunte. La prima è l'area del linguaggio di programmazione e del fatto che questo linguaggio è incompleto e, per certi compiti, difficile da usare (o, come qualcuno afferma, una porcheria). A quel livello (senza divergere dai paradigmi fondamentali di TEX), estensioni come  $\varepsilon$ -TEX, e per certi versi Omega, hanno tentato di semplificare la programmazione fornendo le primitive mancanti che colmassero le ovvie lacune del linguaggio base. Ma, dato che tali funzionalità aggiuntive semplificavano solo la vita al programmatore senza migliorare realmente i risultati della composizione e le funzionalità di (quasi) tutte le nuove primitive potevano essere ottenute con qualche sforzo extra nel linguaggio base, abbiamo deciso di evitarle nello sviluppo di LATEX, dato che il loro uso avrebbe dato come risultato una versione di LATEX che avrebbe funzionato solo su una piccola frazione delle installazioni, senza alcun vantaggio significativo per l'utente. Il LATEX Project, insieme a persone di ConTEXt ed  $\varepsilon$ -TEX, si è veramente sforzato per produrre una definizione di sintassi arricchita per  $\varepsilon$ -TEX CARLISLE (1998) che pensavamo avrebbe fornito abbastanza benefici per passare ad un successore di TEX che implementasse questo set esteso. Tristemente, poco tempo dopo questa proposta, il lavoro su  $\varepsilon$ -TEX si è veramente interrotto e niente di tutto ciò fu mai implementato. Ciononostante, qualcosa è cambiato da allora: la base installata di installazioni

abilitate ad  $\varepsilon$ -TEX è cresciuta oltre la massa critica (largamente grazie a PDFTEX che includeva le estensioni  $\varepsilon$ -TEX) cosicché, qualche tempo fa, il LATEX Project ha annunciato ufficialmente che baserà le future versioni di LATEX su questo set esteso di primitive – ed ha recentemente iniziato a produrre realmente codice facente uso di di queste caratteristiche estese (sebbene ancora solo al di fuori del codice del kernel). In sostanza, non abbiamo mai voluto allontanarci dall'essere in grado di far girare LATEX appena spaccettato su una larga base di interpreti installati e abbiamo ritenuto ciò più importante di un linguaggio di programmazione potenzialmente più semplice o meglio ingegnerizzato che quasi nessuno potrebbe utilizzare<sup>5</sup>. Così, invece di provare semplicemente ad influenzare il linguaggio TEX estendendolo, io e alcuni altri nel progetto LATEX abbiamo lavorato anche dall'interno sviluppando il "linguaggio di programmazione sperimentale LATEX" CARLISLE *et al.* (1997). Questo è stato fatto su molti prototipi (il primo realizzato già nel 1993 o giù di lì). La versione corrente è qualcosa che pensiamo possa essere usato con successo ed abbiamo cominciato a fornire i primi pacchetti pubblici in questo linguaggio LATEX3 PROJECT TEAM.

La seconda area è quella che ha a che fare con i problemi di microtipografia, cioè quelli a cui ero principalmente interessato quando esponevo le deficienze di TEX in MITTELBACH (1990). In questa area il mio coinvolgimento era largamente confinato a dare inizio al lavoro per altri.

La terza area è quella che si interessa delle questioni generalmente aperte e irrisolte della tipografia digitale, come i modelli per la rappresentazione del materiale logico MITTELBACH (1997) e contenuti visuali ROWLEY e MITTELBACH (1997), le trasformazioni tra rappresentazione visuale e logica mediante metodi automatici che forniscano comunque la più alta qualità in accordo ad una metrica definita MITTELBACH e ROWLEY (1992), per dare alcuni esempi. Parte di questa ricerca è capire e codificare le regole tipografiche e sviluppare concetti e algoritmi che possano essere guidati da regole parametrizzate, ad esempio per produrre un posizionamento mobile di alta qualità.

Intravedo di iniziare il mio personale progetto di implementazione a basso livello per migliorare TEX? Quasi certamente no, ma intravedo di essere (di nuovo) coinvolto negli sviluppi correntemente in atto e spero di unire alcuni di questi sviluppi. Se questo avverrà in un progetto come exTEX o

5. Ho imparato che questo è un fattore critico quando abbiamo provato ad introdurre LATEX 2 $\varepsilon$  nel 1994, il quale richiedeva l'installazione dei font codificati T1 (per esempio, font contenenti caratteri con segni diacritici). Il passaggio al nuovo sistema è quasi fallito perché gli utenti statunitensi non vedevano alcun vantaggio in un sistema contenente tutti quei caratteri inutili, necessari solo agli europei. Abbastanza fortunatamente, LATEX 2 $\varepsilon$  aveva altri benefici che hanno prevalso per tutti gli utenti, ma è stato per un pelo.

PDFTEX è decisamente irrilevante. In questo momento ci sono ancora molte questioni irrisolte ed è ancora tempo di esperimenti (che possono avvenire contemporaneamente in diversi progetti), ma un obiettivo importante sarebbe riunire i vari sviluppatori per parlarsi delle loro idee e dei concetti dietro le idee.

*GP: Hai citato due sistemi operativi verso cui è stato portato  $\LaTeX$  e sappiamo che gira su diversi OS liberi e non liberi (sia commerciali che no). Che tipo di sistemi e programmi usi principalmente? E perché?*

FM: Per essere precisi,  $\LaTeX$  è interpretato, dunque non è un software che abbia bisogno di essere portato verso alcun sistema;  $\LaTeX$  gira ovunque  $\TeX$  sia stato portato – e  $\TeX$ , per quanto ne so, è stato portato più o meno su tutti i sistemi operativi mai esistiti (con l'eccezione di qualcosa simile a Palm OS); ad esempio, io l'ho usato su Mainframe, VMS, Unix, Multics e Windows.

Uso sia sistemi operativi liberi che commerciali; molto dipende dall'ambiente e dal compito sottostante. A casa uso principalmente Linux con VMware per accedere ad alcuni programmi disponibili solo per Windows. Sul mio portatile la situazione è inversa: lì gira XP nativo e uso cygwin per un ambiente a linea di comando decente con tutti i vantaggi di un buon sistema Unix. Il mio editor preferito è Emacs, che uso praticamente su ogni piattaforma. Mi piace strutturare le cose usando le mappe mentali e per questo l'unico programma veramente buono che ho trovato è commerciale e gira solo sotto Windows – è una delle ragioni per cui questi giorni uso Windows regolarmente<sup>6</sup>.

Nel mondo professionale, in cui mi guadagno da vivere, il sistema operativo predominante è Windows, e nel mondo dei server troverai varianti commerciali di Unix, ma anche un numero crescente di server Linux.

Quello che uso dipende molto dal lavoro sottostante: per qualcuno, come la gestione di progetti, il mondo Windows offre semplicemente gli strumenti migliori; in altri, il software libero (che gira su OS liberi o commerciali) offre migliore qualità o caratteristiche altrimenti non disponibili. Esempi potrebbero essere Perl, Apache, CVS, subversion e altri.

*DW: Ricordo Knuth dire che scrivere The  $\TeX$ book ha portato a centinaia di modifiche a  $\TeX$ , perché era costretto a spiegare le cose al lettore e, quando non ci riusciva, cambiava il programma. Tu hai scritto una certa quantità di libri e sei stato uno sviluppatore importante. Il tuo lavoro come autore ha influenzato il tuo lavoro come sviluppatore, o viceversa?*

6. Ho recentemente riprovato il programma libero *freemind* ed è stato molto migliorato, per cui la cosa può cambiare ancora.

FM: Penso che Don abbia assolutamente ragione nel fare quest'affermazione: penso che sia estremamente prezioso combinare lo sviluppo del software (in realtà qualunque cosa) con lo scriverne in qualche forma. Provare a spiegare ad altri le funzioni e i concetti dietro una creazione aiuta molto a scoprire se qualcosa funzionerà in pratica o no. Se non puoi spiegarlo, o se la spiegazione risulta orribilmente complicata, allora c'è qualcosa di fondamentale sbagliato nella tua creazione e dovresti tornare al tavolo da disegno.

È molto importante qui non fermarsi alla semplice documentazione delle funzioni o delle voci dei menu (sebbene sia un inizio), ma provare realmente a documentare il ciclo d'uso e le ragioni per cui si farebbero le cose in un modo o in un altro. Abbastanza spesso (sia con software libero che commerciale), si trova solo documentazione rudimentale che racconta che esistono questa e quest'altra caratteristica, ma non spiega mai perché si vorrebbe usare quella caratteristica in prima istanza. Quel tipo di documentazione, sebbene necessaria, non ti aiuta a migliorare il tuo strumento (e, abbastanza spesso, sembra che tali caratteristiche siano state implementate solo perché erano facili da aggiungere, senza fornire alcun beneficio reale).

Quindi sì, documentare le idee e i flussi di lavoro è sempre stato per me parte integrante dello sviluppo e/o del miglioramento del mio software, così come del software altrui. In *The  $\LaTeX$  Companion* MITTELBACH e GOOSSENS (2004), per esempio, buona parte di ciò che descrivo è software sviluppato da altri e il processo di provare a spiegare come usarlo e trovare buoni esempi d'uso porta, in molti casi, a miglioramenti nella sintassi o nelle caratteristiche dopo averne discusso con gli autori.

Di conseguenza, il mio consiglio agli sviluppatori è di provare sempre ad autodocumentare le proprie creazioni, o almeno trovare qualcuno che lo faccia per loro (partendo dall'inizio dello sviluppo!), e valutare le scoperte di questo processo: ne risulteranno notevoli miglioramenti al prodotto.

*DW: Grazie molte, Frank, per aver trovato il tempo di fare quest'intervista con noi. I tuoi approfondimenti su  $\TeX$ ,  $\LaTeX$  e lo sviluppo e la diffusione di sistemi complessi in un ambiente di elaborazione distribuito sono affascinanti. Credo che i lettori dell'Angolo delle Interviste del TUG saranno d'accordo.*

*E grazie a te, Gianluca, per aver accettato di farmi condividere quest'intervista con te. È stato un piacere lavorare con te.*

*GP: Grazie, Frank, per aver dato ai lettori di Free Software Magazine un saggio molto ben spiegato su un importante pezzo di software libero. Inoltre, la tua spiegazione sulla LPPL ha chiarito alcuni punti oscuri di un precedente articolo PIGNALBERI (2005). E grazie a te, Dave, per avermi dato la*

possibilità di fare quest'intervista congiunta, che è molto più interessante di un'intervista "normale".

FM: Grazie, Dave e Gianluca, per aver condotto quest'intervista nel modo in cui lo è stata. Mi è piaciuto vederla svilupparsi domanda dopo domanda; nonostante il tempo che ha richiesto (colpa mia) e il fatto che viviamo distanti, mi è sembrato di fare un'intervista faccia a faccia che, penso, sia come doveva essere.

## Note e licenza

La presente intervista è apparsa in origine su TUG Interview's Corner e Free Software Magazine n° 12. L'articolo è © 2006 degli autori, ed è rilasciato con la seguente licenza: "Sono permesse la copia e la distribuzione *verbatim* del presente articolo con ogni mezzo e senza pagamento di diritti d'autore soltanto se la presente nota è parte integrante dell'articolo". Traduzione a cura di Gianluca Pignalberi.

## Riferimenti bibliografici

LATEX3 PROJECT TEAM. «The LATEX Project CVS Repository». Disponibile su <http://www.latex-project.org/cgi-bin/cvsweb/>.

— (1995). «Modifying LATEX». Disponibile con tutte le distribuzioni LATEX.

— (2003). «The LATEX Project Public License (Version 1.3)». Disponibile su <http://www.latex-project.org/lppl/>.

BROOKS, JR, F. P. (1995). *The Mythical Man-Month; Essays on Software Engineering*. Addison-Wesley, Boston, Massachusetts, 2<sup>a</sup> edizione.

CARLISLE, D. (1998). «Notes on the Oldenburg  $\epsilon$ -TEX/LATEX3/ConTEXt meeting». Disponibile su <http://www.latex-project.org/papers/etex-meeting-notes.pdf>.

CARLISLE, D., ROWLEY, C. e MITTELBAACH, F. (1997). «The LATEX3 Programming Language—a proposed system for TEX macro programming». *TUGboat*, **18** (4), pp. 303–308. Disponibile su <http://www.tug.org/TUGboat/Articles/tb18-4/tb57rowl.pdf>.

MITTELBAACH, F. (1990). «E-TEX: Guidelines for Future TEX Extensions». *TUGboat*, **11** (3), pp. 337–345. Disponibile su <http://www.tug.org/TUGboat/Articles/tb11-3/tb29mitt.pdf>.

— (1997). «Language Information in Structured Documents: Markup and rendering—Concepts

and problems». In *International Symposium on Multilingual Information Processing*. Tsukuba, Japan, pp. 93–104. Articolo invitato. Ripubblicato su *TUGboat* 18(3):199–205, 1997. Disponibile su <http://www.tug.org/TUGboat/Articles/tb18-3/tb56lang.pdf>.

MITTELBAACH, F. e GOOSSENS, M. (2004). *The LATEX Companion. Tools and Techniques for Computer Typesetting*. Addison-Wesley, Boston, Massachusetts, 2<sup>a</sup> edizione. With Johannes Braams, David Carlisle, and Chris Rowley.

MITTELBAACH, F. e ROWLEY, C. (1992). «The pursuit of quality: How can automated typesetting achieve the highest standards of craft typography?» In *EP92—Proceedings of Electronic Publishing, '92, International Conference on Electronic Publishing, Document Manipulation, and Typography, Swiss Federal Institute of Technology, Lausanne, Switzerland, April 7–10, 1992*, a cura di C. VANOIRBEEK e G. CORAY. Cambridge University Press, New York, pp. 261–273.

PIGNALBERI, G. (2005). «The LATEX Project Public License». *Free Software Magazine*, (7), pp. 52–54. Disponibile su [http://www.freesoftwaremagazine.com/free\\_issues/issue\\_07/tex\\_license/](http://www.freesoftwaremagazine.com/free_issues/issue_07/tex_license/).

ROWLEY, C. A. e MITTELBAACH, F. (1997). «Application-independent representation of multilingual text». In *Europe, Software + the Internet: Going Global with Unicode: Tenth International Unicode Conference, March 10–12, 1997, Mainz, Germany*, a cura di UNICODE CONSORTIUM. The Unicode Consortium, San Jose, CA. Disponibile su <http://www.latex-project.org/papers/unicode5.pdf>.

ZIEGLER, J. (1994). «Technical report on math font encoding (version 2)». Technical report, LATEX3 project. Disponibile su CTAN: [info/1tx3pub/13d007.\\*](http://ctan.org/info/1tx3pub/13d007.*).

- ▷ Frank Mittelbach  
LATEX Project  
[frank.mittelbach@latex-project.org](mailto:frank.mittelbach@latex-project.org)
- ▷ Gianluca Pignalberi  
Free Software Magazine  
[g.pignalberi@freesoftwaremagazine.com](mailto:g.pignalberi@freesoftwaremagazine.com)
- ▷ Dave Walden  
TUG  
[dave@walden-family.com](mailto:dave@walden-family.com)