



Dr. Gabriele Floria DDS

VJO editor

Original Article
Published on 15-12-00

La valutazione del software per personal computer in uno studio ortodontico

Introduzione

L'utilizzo del computer in uno studio ortodontico è ormai molto comune, la videoscrittura, l'archiviazione delle informazioni contabili ed anagrafiche, la gestione delle immagini radiografiche e fotografiche, e le analisi cefalometriche, trovano infatti in esso l'indispensabile supporto tecnologico. Come ogni altro strumento del nostro studio deve essere perciò valutato per rapporto costo/beneficio, ma la particolarità del mezzo in questione impone conoscenze hardware e software per ottenere un binomio efficiente e vantaggioso in funzione delle specifiche esigenze.

L'investimento economico effettuato per allestire procedure gestionali od operative rischia di subire una netta perdita, nel caso in cui si decida di cambiare in pochi anni sistema operativo o piattaforma informatica. Si impone quindi una riflessione attenta per effettuare scelte consapevoli, ragionate e non condizionate da operazioni di marketing che in ultima analisi si rivelino uno svantaggio nel medio lungo periodo.

La scelta del sistema operativo per i personal computer di uno studio ortodontico viene spesso trascurata dagli utenti non esperti, in realtà considerando la scelta dell'applicativo come prioritaria ed adeguandoci alle specifiche richieste, spesso si rischia di perdere di vista il binomio processore centrale-sistema operativo che rappresenta il reale "motore" del nostro computer e con esso anche molte caratteristiche, che influenzeranno la totalità dell'hardware e del software futuro.

Definizione di sistema operativo:

Il sistema operativo di un computer è quel particolare software che la macchina carica alla sua accensione e che consente l'interazione dei vari dispositivi hardware con la CPU (Central Processing Unit), inoltre fornisce servizi e supporto agli applicativi (3).

Schematizzando possiamo distinguere tre aspetti di questo:



Sarebbe anche interessante affrontare il concetto di "free software" che sta dietro a questo sistema operativo e a molto altro software di ottima qualità, spesso sottostimato, ma questo aprirebbe a speculazioni filosofiche, economiche ed anche politiche che richiederebbero troppo spazio e necessiterebbero diversa collocazione.

Questo sistema operativo è adatto ai cosiddetti "power user", cioè coloro i quali hanno conoscenze tecnico-informatiche di un certo livello, supponendo la necessità di utilizzare caratteristiche di base del sistema operativo, quali ricompilazione del kernel e compilazione di applicativi autoprodotti, ma è adatto anche all'utente normale che desidera un sistema operativo compilato su misura per il proprio hardware che interagisca con il computer tramite un applicativo specifico, come un programma cefalometrico od un gestionale pazienti e immagini, perché è in grado di fornire un sistema gerarchico ben controllabile e configurabile per livelli (2). È adatto inoltre a chi debba gestire una rete di computer medio-grande perché la gratuità delle licenze consente di risparmiare ed avere un prodotto finale con un alto grado di stabilità e sicurezza informatica. Gli svantaggi nel settore ortodontico consistono nella difficoltà di reperire software dedicato perché la gran parte delle software house negli ultimi anni si è dedicata allo sviluppo di programmi per altri sistemi operativi.

Programmi di utilità ovvero il software

I programmi di utilità sono un insieme di piccole applicazioni utili per la gestione pratica del sistema. Teoricamente, tutte le funzionalità amministrative per la gestione del sistema potrebbero essere incorporate in una shell, in pratica, questo solitamente non è. Dal momento che le shell tradizionali incorporano alcuni comandi di uso frequente, spesso si perde la cognizione della differenza che c'è tra le funzionalità fornite dalla shell e i programmi di utilità.

1. il kernel
2. la shell
3. i programmi di utilità

Kernel

Il termine kernel deriva dal tedesco "kern" che significa nucleo (ma anche anima in senso tecnico) ed infatti il kernel è il cuore del sistema, rappresenta il livello più basso con il quale ogni programma deve interagire per qualunque operazione, infatti esso controlla gli input e gli output (il flusso dei dati) da e per le periferiche, alloca le risorse in funzione delle esigenze, e provvede alla gestione dello spazio per la memorizzazione dei dati. Svolge quindi un ruolo chiave ed il suo corretto funzionamento e condiziona pesantemente tutto il sistema hardware-software. Inoltre kernel diversi ma con funzionalità compatibili favoriscono quella caratteristica definibile come **portabilità**, ovvero la possibilità di trasferire gli stessi programmi su piattaforme differenti (1,4).

Shell

Il kernel è stato spesso descritto come una "perla" e come tale non può essere disgiunto dalla "conchiglia", ovvero dalla shell. Quest'ultima avvolge il kernel consentendo all'utente di interagire col sistema. La shell può avere diverse forme, dalla più semplice riga di comando con un prompt, come quella del DOS, ad un'interfaccia grafica ad icone o da un sistema di riconoscimento a comando vocale. Si tratta sempre di shell.

Scegliere il Sistema operativo

MS-DOS

È un sistema operativo ormai datato ma che conserva la caratteristica di girare su processori Intel 286 e superiori, 640 Kb di memoria sono sufficienti per il suo funzionamento. Non è multi-user (cioè non supporta più utenti sulla stessa macchina), né multitasking (non può eseguire più programmi nello stesso momento), né multithreading (non può eseguire differenti parti di programma contemporaneamente). Inoltre MS-DOS è in grado di interagire pochissimo con altri sistemi operativi, ha grossi limiti per quanto riguarda la gestione della memoria, non comprende programmi che consentano il networking (il lavoro in rete informatica), né utility per lo sviluppo (quali compilatori e debugger).

Windows 95/98

È preinstallato (e quindi pre-pagato) sulla maggioranza dei personal computer in commercio, è noto a molti utenti (per cui è semplice trovare aiuto quando ci sono problemi), è usato anche da moltissimi programmatori (per cui è facile farsi fare programmi per esigenze specifiche). Ha un'interfaccia grafica intuitiva. Consente un

Quindi il sistema operativo nella sua totalità si occupa principalmente di: controllare il flusso dei dati, allocare le risorse, organizzare e tenere traccia delle memorizzazioni dei dati, mantenere il giusto livello di sicurezza, e controllare l'assenza di malfunzionamenti dell'insieme.

Queste funzioni importantissime sono praticamente invisibili, perché l'interfaccia utente è quasi sempre un programma applicativo che ci consente di memorizzare una cartella clinica, un tracciato cefalometrico, ovvero tutto ciò che soddisfa i nostri bisogni di utilizzatori finali. Talvolta si attribuiscono responsabilità al programmatore che ha realizzato l'applicativo che non funziona correttamente, e che in realtà sono difetti dell'hardware o del sistema operativo, o ancora sono derivanti dalla non felicissima unione hardware-software.

Valutazione della qualità del software

Uno studio ortodontico necessita di diversi programmi: un gestionale per la fatturazione e gli adempimenti burocratici (di giorno in giorno sempre maggiori), un software cefalometrico, un database con le cartelle cliniche dei pazienti, un archivio immagini e radiografie digitalizzate.

Se il problema della informatizzazione dello studio non viene affrontato in modo globale, difficilmente i singoli programmi acquistati in tempi diversi e da differenti produttori, saranno in grado di interscambiarsi i dati a comune e di aggiornarsi automaticamente. Il problema sembrerà piccolo per chi deve gestire un basso numero di informazioni ma diviene enorme all'aumentare del flusso dei dati. Per questo motivo lo schema da seguire, secondo il mio modesto parere, consiste nell'identificare la piattaforma (Microsoft, Apple Macintosh, Unix, BeOS etc...) che si desidera utilizzare in funzione delle esigenze, per poi cercare sul mercato ciò che soddisfa i nostri desiderata. Nel caso non si riesca a trovare qualcosa che sia di nostro gradimento si può richiedere ad un programmatore la realizzazione di un software su misura per le nostre esigenze. In entrambi i casi dobbiamo essere in grado di valutare qualitativamente un software. Identifichiamo quindi delle caratteristiche, alcune delle quali sono per così dire interne (verificabilità e riusabilità) mentre le altre possono essere percepite dall'utente finale (2).

- Affidabilità
- Correttezza
- Robustezza
- Usabilità
- Verificabilità
- Manutenibilità
- Riusabilità
- Comprensibilità
- Interoperabilità
- Affidabilità

Affidabilità

È la corrispondenza del programma con i suoi requisiti. Qualsiasi macchina, compreso il computer, è affidabile se si comporta esattamente come ci si aspetta che si comporti. Può sembrare banale ma la piena affidabilità nel creare un software richiede un attento lavoro d'equipe da parte di utenti finali con i programmatori per identificare le esigenze, ed una successiva fase di "debugging" ovvero di rimozione degli errori che inevitabilmente ci saranno nelle versioni preliminari.

Correttezza

Può genericamente rientrare nell'affidabilità, ma precisamente con questo termine si identifica la corrispondenza del programma con le sue specifiche. Per chiarire meglio diciamo che i requisiti sono ciò che l'utente desidera veramente, mentre le specifiche sono ciò che i programmatori credono l'utente desideri.

È un punto molto importante, perché solo chi ha competenze come

collegamento in rete con altri PC che siano dotati dello stesso sistema operativo, pur adattandosi meglio, secondo il parere dell'autore al personal computer singolo. E' suscettibile di attacco da parte di virus informatici che possono causare la perdita di dati e la distruzione del sistema operativo; necessita quindi di sistemi di protezione (software antivirus) costantemente aggiornati.

Le differenze tra i due sistemi, 95 e 98, sono apparentemente limitate in quanto l'utente percepisce solo un lieve miglioramento delle prestazioni, ma in realtà il kernel di Windows 98, evoluzione del 95, ha subito migliorie in stabilità e gestione della memoria.

Windows NT 4

Windows NT e' disponibile solo su Digital Alpha, e processori Intel x86. E' multiuser, ha capacita' di multitasking e multithreading. La sua interfaccia grafica è molto simile a Windows 95/98 e questo rende più agevole il passaggio dall'uno all'altro. Essendo stato progettato per un uso in rete, infatti NT significa network, non dispone della stessa facilità di configurazione e dello stesso numero di driver di Windows 95/98.

Apple Macintosh

Il sistema operativo di Apple per i Macintosh funziona solo sui Macintosh. Inoltre soffre della mancanza di strumenti di sviluppo, di scarsa interoperabilità con altri sistemi operativi, di un multitasking primitivo, assenza di multi-threading. Le ultime versioni del System 9 hanno risolto molti dei problemi aumentando le capacità di multitasking e l'affidabilità dell'intero sistema. Anche le capacità di networking dei Mac sono migliorate, rendendo questa piattaforma probabilmente la migliore workstation per grafica e desk-top-publishing per utenze non professionali. Purtroppo i programmi applicativi per questi sistemi operativi sono scarsi in campo ortodontico, anche se di recente si sono riaffacciati molti programmatori americani che hanno realizzato ottimi software gestionali sfruttando la caratteristica facilità di utilizzo di questi computer.

Solaris 2.6

Solaris e' un ottimo sistema operativo UNIX, a detta di molti esperti è il migliore in circolazione, per stabilità ed affidabilità.

Ha costi piuttosto elevati e gira su piattaforme Sparc e Intel. E' multitasking, multiuser, multiplatform, e multithreading. Viene fornito preassemblato nei computer SUN che in genere coprono compiti di calcolo matematico-statistico o per la ricerca scientifica, ma è acquistabile anche separatamente. Richiede una conoscenza di alto livello tecnico e possiede una interfaccia grafica piuttosto essenziale. Recentemente sono stati rilasciati i codici sorgente di questo sistema.

OS/2

E' stato il primo sistema operativo a diffusione di massa ad avere un'interfaccia grafica orientata agli oggetti, anticipando nelle edizioni 2.x caratteristiche

utente e come programmatore può formulare le migliori specifiche dei requisiti, infatti molto spesso l'utente pensa alle sue necessità operative, mentre il programmatore pensa agli algoritmi logici per realizzare il programma e non sempre i colloqui si traducono nel migliore dei compromessi.

Robustezza

E' la capacità del programma di comportarsi in modo affidabile anche in presenza di situazioni anomale, dettate per esempio da errori umani o altre circostanze esterne, non deve essere confuso con il concetto di "user-friendly" ovvero con la usabilità, perché non riguarda solo l'interfaccia. Un buon programma deve prevenire per esempio immissioni errate in campi di database o correggerli automaticamente prima della loro immissione..

Usabilità

E' la capacità di un programma di rendersi facilmente comprensibile agli occhi dell'utente finale. E' un requisito tanto più importante tanto più inesperto è l'utente finale. Non dobbiamo stupirci se molti power-user preferiscono un'interfaccia testuale alle sofisticate icone dalla grafica sempre più avida di requisiti hardware. I programmatori dicono che "l'intelligenza dell'interfaccia", sempre che così la si possa definire, deve essere inversamente proporzionale a quella dell'utilizzatore finale, per indicare che a soggetti inesperti devono corrispondere programmi a prova di errore e viceversa.

Verificabilità

E' la possibilità di definire un algoritmo che permetta di capire se il programma è corretto. Esistono diverse categorie di programmi, quelli matematici per esempio sono facilmente verificabili all'opposto i database non lo sono. Questo permette di fornire all'utente finale un prodotto verificato.

Manutenibilità

Spesso i software devono essere aggiornati per il mutare delle esigenze o per abilitare nuove funzioni. Con questa caratteristica si identifica la facilità con la quale si può intervenire per tali modifiche.

Riusabilità

E' la possibilità di riutilizzare parti di programma all'interno di un'altra applicazione.

Comprensibilità

E' data dal grado di leggibilità del codice e dalla documentazione del programma, ovvero dalla possibilità di comprensione da parte di chi non ha scritto quel programma.

Interoperabilità

E' la capacità di un programma di interagire con il software e l'hardware che lo circonda. Riguarda lo scambio dei dati ed è molto importante per l'utente finale.

Il ciclo "vitale" del software

Al pari di altri prodotti commerciali anche il software ha un suo ciclo per così "vitale" ovvero è caratterizzato da una serie di eventi che ne identificano la nascita, l'uso e la dismissione (5).

Fra i vari modelli che sono stati proposti emerge, per logicità e comprensibilità, quello cosiddetto "a cascata" che prevede una serie di eventi consecutivi che forniscano un "output" al termine di ogni fase che rappresenti "l'input" per la fase successiva.

Nel dettaglio queste fasi sono:

Raccolta dei requisiti: E' in sostanza un elenco delle funzionalità che dovranno essere implementate. Escludendo l'eventualità che lo sviluppatore e l'utente siano la stessa persona, questa fase prevede una serie di interviste rivolte agli utenti, per individuare ciò che il programma deve fare. Come abbiamo visto la corrispondenza e la conoscenza parziale dei due specifici campi di lavoro, nel nostro caso quello ortodontico e quello informatico, rappresenta l'elemento ponte che consente la realizzazione di un buon prodotto. Non è un caso che le software house che si dedicano ad uno specifico settore industriale o professionale riescano maggiormente ad accontentare l'utenza finale

che avrebbero acquisito popolarità con Windows95. Pur avendo un discreto numero di affezionati, non riscuote di molta popolarità. Il problema più grosso di questo sistema operativo è che la casa produttrice, la IBM lo ha sostanzialmente abbandonato in favore di Windows NT.

BeOS 4.5

È l'ultimo arrivato sul mercato e dalla sua nascita (1995) è stato concepito per gestire grandi flussi simultanei di dati audio, video e grafica bi e tri-dimensionale. Nasce con l'obiettivo dichiarato di lasciare libere quante più risorse possibili per gli applicativi, inoltre è in grado di gestire più processori sullo stesso computer e vanta un filesystem a 64 bit (l'unico ad oggi per PC). Queste sue prerogative lo rendono molto potente nel multithread e nel multitasking. È in grado di gestire anche filesystem diversi (MacOS, HFS, FAT16/32, vFat, linux ext2 etc...) ed ha tutti gli strumenti per l'utilizzo in rete intranet che internet, ma sembra non molto adatto all'utilizzo multi-utente. Il suo principale difetto pare essere la sua giovinezza e la relativa mancanza di applicativi, ma essendo dotato di ottime caratteristiche pare essere una stella nascente nel panorama informatico mondiale.

Linux

Ogni sistema operativo ha il suo kernel, ma nel caso di questo sistema operativo questa affermazione è vera solo in parte, infatti Linux consente la ricompilazione (1) del kernel in funzione delle caratteristiche hardware o delle specifiche esigenze dell'utente. Possiamo quindi affermare che ogni singola macchina ha un suo proprio kernel creato su misura che ne conferisce dinamicità, e rappresenta secondo l'autore una valida risposta al tumultuoso proliferare di periferiche e nuovi standards che appesantiscono i processi di calcolo e costringono i sistemi operativi statici (non modificabili) ad avere giganteschi codici macchina. È necessario spendere altre due parole sul concetto di dinamicità: Nel processo di ricompilazione è possibile scegliere se avere un kernel monolitico oppure modulare. La struttura monolitica viene caricata in memoria all'avvio della macchina, nella sua interezza, mentre il kernel modulare accede ai moduli, porzioni di kernel pronte all'uso, solo quando necessario. Questo conferisce fluidità al sistema ottimizzando la gestione delle risorse hardware. Le prime versioni sono state sviluppate interamente da Linus Torvalds presso l'Università di Helsinki in Finlandia, ma ciò che rende Linux così diverso è il fatto di essere cresciuto grazie all'aiuto di moltissimi ed eterogenei gruppi di programmatori UNIX ed esperti di Sistemi Operativi che hanno messo a disposizione il codice da loro prodotto, gratuitamente. Questa eterogeneità si riferisce sia alle competenze tecniche che al settore di provenienza nonché alla dislocazione geografica. Affinché queste comunità di programmatori potessero lavorare insieme era necessario un efficiente strumento di comunicazione. Lo strumento

realizzando ottimi prodotti di nicchia.

Studio di fattibilità: In tale fase si analizzano obiettivi, tempi e costi. Se un prodotto per essere ultimato richiede tempi troppo lunghi o costi troppo elevati si trasforma automaticamente in un fallimento commerciale perché risulta superato o fuori mercato. Si definisce spesso tale fase col termine di "make or buy" perché talvolta è conveniente acquistare ciò che sarebbe troppo costoso produrre. Naturalmente è necessario verificare la percentuale di corrispondenza tra le esigenze dell'utente e le specifiche del prodotto acquistabile sul mercato.

Analisi dei requisiti: L'analisi dei requisiti viene formalizzata tramite diagrammi di flusso o gli schemi entità-relazione al fine di individuare passo passo che cosa deve fare l'applicazione. In questa fase non si deve ipotizzare come realizzare concretamente il requisito per non pregiudicare l'obiettivo finale. Si deve in altri termini solo delineare un documento che va sotto il nome di "specificazione dei requisiti".

Progettazione: Nella fase di progettazione di un software ci si preoccupa del come si possano soddisfare i requisiti elencati precedentemente. Senza entrare nello specifico tecnico basti sapere che si identifica una serie di moduli software elencandone le funzionalità di ognuno. L'insieme di questi moduli e delle loro relazioni forma l'architettura dell'applicazione. Più tali moduli risultano indipendenti tra loro e più sarà facile modificare in futuro il software. **Implementazione:** Si esegue la scrittura del codice, se le fasi precedenti sono state ben eseguite non si dovrebbero richiedere modifiche ulteriori che provocherebbero solo un rallentamento della programmazione, un aggravio di costi, ed un software molto più difficile da modificare in seguito.

Manutenzione: Sono tutte quelle operazioni che vengono eseguite dopo la consegna e l'installazione del software. La richiesta di modifiche è soggetta all'utilizzo specifico che l'applicativo deve svolgere e potrà di conseguenza avere costi e tempi variabili ma non è infrequente riscontrare costi per manutenzione pari al 70% del complessivo sviluppo.

Conclusioni

La scelta del software gestionale di studio, di analisi cefalometrica, di acquisizione immagini o altro deve essere subordinata alle 9 caratteristiche identificanti i criteri qualitativi sopra citati, e deve essere strettamente correlata con le esigenze operative del singolo professionista, privilegiando per esempio sistemi operativi più efficienti per la grafica o per la condivisione in rete delle risorse informatiche, ma deve anche tener conto del rapporto costo/beneficio e last but not least, della prospettiva futura che il settore della tecnologia informatica seguirà nei prossimi anni, e alla quale anche l'ortodontista dovrà adeguarsi per investimenti e conoscenza tecnica.

Bibliografia:

1. Cavaliere F. La compilazione del kernel. Inter-punto-net 1999; 48:73-78.
2. Floria G, Vergari A, Xenakis D. The first on-line orthodontic journal: an international experience. In: Atti del II World Congress on Biomedical Communications; Academic Medical Center University of Amsterdam The Netherlands; AMC 1999. P.51.
3. Parsons J., Oja D. Computer concepts. International Thomson Publishing company; USA 1996.
4. Rubini A, Zanetti G. Il sistema operativo Linux. Login: Building the information highway 1996;1:46-52.
5. Vandoni L. Ingegneria del software. Dev Infomedia1998; 57:81-83

fu Internet ed essendo Linux il sistema operativo scelto da queste persone, significa che gli strumenti e le utility necessarie ad utilizzare la Rete furono tra le prime a comparire. Oltre ad essere sviluppate appositamente per Linux, molte delle applicazioni furono per così dire "importate" da quello che era il miglior software disponibile in quel momento sul mercato (2).

Le caratteristiche di Linux: E' multitasking, è multiuser, è multiplatform. Ha una caratteristica chiamata "shared copy-on-write pages among executables" ovvero i processi multipli possono utilizzare la stessa memoria per girare. Quando uno di questi cerca di scrivere nella parte di memoria condivisa, quella pagina (4KB di memoria) viene copiata da qualche altra parte. Copy-on-write ha due vantaggi: maggiori prestazioni e riduzione del consumo di memoria.

Ha una gestione della memoria virtuale attraverso la paginazione (senza cioè dover scaricare l'intero processo) sul disco: su una partizione separata, su un file all'interno del filesystem, o entrambi, con la possibilità di aggiungere al volo - qualora ce ne fosse bisogno - ulteriori aree di swap (senza dover riavviare la macchina o interrompere il processo). Un totale di 16 aree di swap da 128MB possono essere utilizzate contemporaneamente per un teorico ammontare di 2 GB di spazio di swap utilizzabile.

Utilizza un pool di memoria unificato per i programmi e la memoria cache in maniera tale che tutta la memoria libera possa essere utilizzata come cache, e al tempo stesso la cache possa essere ridotta qualora ci sia bisogno di lanciare un grosso programma. Ha la completa disponibilità dei sorgenti, compreso l'intero kernel e tutti i driver, gli strumenti di sviluppo e tutti i programmi utente; inoltre è tutto liberamente distribuibile (4).

Dispone di pseudoterminali e console virtuali multiple: è possibile effettuare diversi login indipendenti dalla stessa postazione e passare da una all'altra con una semplice combinazione di tasti (spesso ALT-F1 - ALT-F2...). È possibile avere fino a 64 sessioni parallele, contemporanee e indipendenti.

È supporto nativo per molteplici filesystem: minix-1, Xenix System V, MS-DOS, HPFS (OS/2 2.x), VFAT (Windows 95/98), NT, HFS (Apple Mac), FFS (Amiga), CD-ROM, NFS...

Il suo filesystem nativo supporta partizioni fino a 4TB e nomi lunghi fino a 255 caratteri. Ha un networking TCP/IP nativo (comprendente ftp, telnet, NFS, etc.)

Può agire da server per reti AppleTalk, da server per reti Microsoft (emulando LanManager, NT...) e da client (WfWg, W95, NT). Può agire da client o server in una rete Novell Netware.

Linux si basa sullo standard POSIX per i sistemi operativi che è stato derivato originariamente dalle funzionalità del mondo UNIX.

Da rilevare il fatto che UNIX è compatibile con Linux a livello di chiamate di sistema, il che significa che la maggior parte dei programmi scritti per altre versioni di UNIX o per Linux possono essere compilate e funzionare anche sugli altri sistemi con

poche o spesso nessuna modifica. Generalmente Linux gira molto più velocemente delle altre versioni di UNIX sullo stesso hardware. L'architettura Linux è già la seconda piattaforma più comune come Web Server negli USA seconda solo a Solaris (utilizzando entrambi l'Apache Web Server). Tutti i sistemi basati sull'architettura UNIX (linux incluso) sono praticamente immuni da virus informatici perché pur essendo possibile la loro compilazione (il concetto di virus è nato sotto UNIX) la gestione dei permessi sui singoli file impedisce, se la rete è ben strutturata, la trasmissione da un utente all'altro e quindi al massimo potrà distruggere i dati dell'utente malcapitato o imprudente.

To cite this article please write:

Floria G. La valutazione del software per personal computer in uno studio ortodontico. Virtual Journal of Orthodontics [serial online] 2000 Dec 15; 3(3):[5 screens] Available from URL: <http://www.vjo.it/033/comport.htm>

[about us](#) | [current issue](#) | [home](#)

Virtual Journal of Orthodontics ISSN - 1128 6547
Issue 3.3 - 2000 - <http://www.vjo.it/vjo033.htm>
Copyright © 1996-2000 All rights reserved
E-mail: staff@vjo.it