ISSN: 0394-0705

ACCADEMIA NAZIONALE DEI LINCEI

ANNO CCCXC - 1993

CONTRIBUTI DEL
CENTRO LINCEO INTERDISCIPLINARE
«BENIAMINO SEGRE»
N. 87

SEMINARIO

DISCIPLINE UMANISTICHE E INFORMATICA

IL PROBLEMA DELL'INTEGRAZIONE

(Roma, 8 ottobre 1991)

a cura di Tito Orlandi

ROMA
ACCADEMIA NAZIONALE DEI LINCEI
1993

Il presente volume pubblicato grazie al contributo della «Associazione Amici della Accademia dei Lincei»

Finito di stampare nel mese di giugno 1993

Stampa: Tipografia «STI», Via Sesto Celere, 3 - 00152 Roma

TITO ORLANDI

INTRODUZIONE

I contributi che presentiamo in questo libro sono stati letti nel Seminario intitolato a *Il problema dell'integrazione*, tenuto il giorno 5 ottobre 1991 presso l'Accademia dei Lincei.[1] Il Comitato Scientifico era costituito dai Soci Sabatino Moscati (allora Direttore del Centro), Ignazio Baldelli e Tito Orlandi. Il Comitato Organizzativo era costituito da Tito Orlandi (coordinatore), Giovanni Adamo, Giovanni Azzena, Giuseppe Gigliozzi, Raul Mordenti, Paola Moscati, Manuela Tascio.

Il Seminario ha costituito la prima manifestazione nell'ambito di una ricerca condotta dal Centro Linceo Interdisciplinare Beniamino Segre, relativa a *Le discipline umanistiche a confronto coi metodi e le tecniche dell'informatica: Continuità della tradizione e rinnovamento strutturale.* Il Centro Linceo, che «si propone di sviluppare il pensiero matematico nel senso più ampio, in rapporto a tutte le Scienze», è stato attento fin dalla sua istituzione a cogliere quanto si veniva elaborando nel campo delle applicazioni umanistiche dei calcolatori elettronici, e ne ha promosso l'evoluzione organizzando conferenze e altre attività[2] e concedendo apposite borse di studio.

Più recentemente il Centro Linceo ha definito una attività di ricerca che, dando per scontati gli aspetti tecnici e settoriali dell'applicazione dell'informatica alle discipline umanistiche, è rivolta agli aspetti del rinnovamento metodologico, prodotto dall'incontro fra le consolidate metodologie tradizionali della matematica e delle singole discipline umanistiche. Questa ricerca non ha come fine quello di ampliare i risultati di qualche applicazione particolare dell'informatica in ambito umanistico, ma di arrivare al cuore dei problemi che questo tipo di applicazione comporta.

I rapporti fra alcune delle discipline umanistiche e l'informatica,

---

1. I contributi di Amilcare Bietti (Metodi matematico-statistici per la preistoria e la paletnologia) e di Mirella Casini Schaerf (Strumenti per il parsing di testi in linguaggio naturale) non sono stati inviati per la pubblicazione.

2. Cf. i «Contributi» del Centro nn. 13, 17, 19, 28, 47, 56, 61.

termine con cui intendiamo la disciplina che studia il trattamento automatico dell'informazione, sono stati fin dall'inizio molteplici, e fecondi di risultati scientifici per l'una e l'altra parte. È noto come gli studi di logica formale e di linguistica siano stati e continuino ad essere oggetto di attenta considerazione da parte di chi si occupa di linguaggi di programmazione; e come alcuni linguaggi di programmazione (prolog, lisp, etc.) siano particolarmente adatti a ricerche teoriche nel campo della logica e dell'organizzazione della conoscenza.

Differente appare la situazione dei rapporti fra l'informatica e gli specifici ambiti di ricerca delle discipline umanistiche in generale. In questo caso si è generalmente fermata l'attenzione ai lati puramente applicativi e tecnologici, trascurando di considerare nella giusta misura il rapporto di tipo metodologico fra i due settori. D'altra parte si è spesso confusa la ricaduta pratica di alcune attività umanistiche (tecniche museali, enciclopedie elettroniche, procedimenti per la stampa) con la vera e propria attività di ricerca a livello scientifico. Di conseguenza non si è ancora stabilito un corretto rapporto di piena comprensione fra gli specialisti di informatica e gli studiosi, a cui vengono offerti soprattutto pacchetti applicativi che per quanto potenti, e dunque utili sotto molti aspetti, non risultano adatti a soddisfare le esigenze sofisticate proprie delle discipline umanistiche a livello di ricerca.

Si lamenta in particolare la mancanza di flessibilità e specificità, nei riguardi sia dell'aderenza ad una documentazione di tipo particolare, sia del trattamento di materiali scritti in alfabeti estranei alla normale esperienza informatica o che necessitano di strumenti descrittivi molto sofisticati. In alternativa, viene offerta una collaborazione tecnica diretta, che crea programmi «su misura» per alcune esigenze. Essi tuttavia si rivelano presto inadeguati al sopravvenire di progressi e approfondimenti negli studi. Nel complesso si riscontra, da parte degli specialisti di informatica, una comprensione non buona di quale sia il reale, concreto e minuto lavoro di ricerca nel campo delle discipline umanistiche. Questa incomprensione tocca anche e specialmente il livello metodologico di tale lavoro, e il rapporto in tale ambito fra la raccolta e il trattamento della documentazione (su questo versante molto è stato fatto per venire incontro alle più banali esigenze degli studiosi) e la sua valutazione e sistemazione scientifica. Anche in quest'ultimo campo l'informatica può entrare con risultati importanti.

Nella prassi, attualmente le ricerche umanistiche si giovano soprattutto dei pacchetti applicativi preparati per la cosiddetta «office automation» (trattamento di testi, gestione di banche dati, pacchetti statistici, grafica, stampa) e solo in alcuni casi di pacchetti più specifici, come quelli per le concordanze. Fra questi pacchetti, sono assai rari quelli disponibili per gli studiosi in generale, mentre spesso essi sono prodotti all'interno di una singola ricerca, e risultano inidonei per l'uso in ricerche anche affini. Lo stesso vale per i pacchetti per la gestione delle immagini e del suono, che sono rivolti piuttosto alle necessità progettuali e creative che non a quelle di analisi e dunque di ricerca.

Soprattutto è da lamentare la scarsa considerazione per la necessità di un ambiente unitario nel quale possa essere agevolmente organizzato lo scambio di documentazione fra i diversi studiosi, e l'utilizzazione interattiva di diversi pacchetti o strumenti applicativi da parte di uno studioso. Si nota infatti la tendenza a considerare a sè ciascuna delle esigenze particolari, trascurando il complessivo e multiforme lavoro di ricerca.

Questo ambiente unitario, che deve essere adatto al lavoro umanistico, può risolvere anche un problema di ordine più pratico, ma non meno pressante. Esso riguarda la quantità di dati che ogni studioso vorrebbe avere già a disposizione quando inizia un lavoro, e che invece richiedono moltissimo tempo e grande cura per essere convenientemente memorizzati. Parlo di corpora testuali nelle diverse lingue antiche e moderne, e archivi di notizie e documenti storici, storico-letterari, storico-artistici, etc.

Le soluzione sono in sostanza tre: prima di tutto grossi centri specializzati, che in effetti esistono, ma non sono sufficienti a coprire, neppure in prospettiva, le necessità esistenti, e comunque presentano alcuni rischi di accentramento. D'altra parte i sistemi di memorizzazione automatica, in cui molti ripongono fiducia, si sono rivelati inefficaci, perché si è visto che la riduzione di un testo in machine readable form è cosa assai diversa dalla sua pura digitalizzazione. Occorre allora che si formi una vastissima rete, in cui i singoli studiosi e le loro équipe, che sono grandi produttori di dati, se li possano scambiare e gestire con gli stessi strumenti software. Questo presuppone l'identificazione di un ambiente operativo uniforme, e delle regole minimali affinché i dati siano scambiabili.

Il progetto di ricerca che il Centro Interdisciplinare ha deciso di

promuovere si propone di chiarire e rendere esplicita questa situazione, e di indicare le condizioni per migliorarla, affrontando in modo nuovo e globale i problemi dell'automazione in campo umanistico. I campi da esplorare sono parecchi, ma il lavoro è già stato iniziato anche da altri ricercatori in Italia e all'estero, coi quali ci si propone di stabilire una fattiva collaborazione.

In primo luogo si devono assumere le caratteristiche intrinseche dell'informatica e dei suoi procedimenti, confrontandole con le metodologie specifiche delle discipline umanistiche, come si sono venute formando attraverso la loro secolare tradizione. Inoltre occorre approfondire i problemi relativi al rapporto fra i linguaggi formali o artificiali, ed i linguaggi naturali, e la reale possibilità di analizzare gli uni mediante gli strumenti ideati per gli altri. Sotto questo aspetto, assumono grande rilievo i problemi relativi alla codifica, intesa come formalizzazione delle rappresentazioni, sia in linguaggio naturale che come oggetti. Di qui i problemi relativi al rapporto fra l'organizzazione strutturale di una banca dati, e la struttura della realtà storica che essa è chiamata a rappresentare. Nel complesso, si tratta di chiarire i rapporti fra la parte formalizzabile delle metodologie umanistiche, e i procedimenti automatici applicabili a tali discipline.

La riflessione su questi problemi teorici sarà accompagnata da sperimentazioni relative anche ai procedimenti pratici, minuti e quotidiani degli studiosi nell'ambito delle proprie ricerche, per le quali si vengono approntando le necessarie strutture. In particolare sembra necessario:

1) Chiarire i diversi piani in cui si collocano le attività umanistiche, distinguendo quello della ricerca da quello della diffusione o dell'insegnamento del patrimonio delle conoscenze già acquisite (linguistiche, artistiche, storiche, archeologiche, etc.).

2) Chiarire quale sia il terreno comune fra informatica e discipline umanistiche, al di là delle ovvie differenze di metodo e di prassi. Solo partendo dall'attenta considerazione di questo terreno comune, è possibile stabilire una corretta collaborazione fra gli specialisti dei due settori.

3) Individuare i settori nei quali è maggiormente possibile e consigliabile l'applicazione informatica in ambito umanistico.

4) Individuare i principali problemi di ricerca che è opportuno aiutare a risolvere mediante procedimenti automatici.

5) Suggerire le linee di sviluppo in questa direzione, in base alle

esperienze fatte e ai relativi risultati, approfondendo i lati positivi e negativi di tali esperienze.

6) Richiamare l'interesse degli studiosi di discipline umanistiche, che non abbiano fatto ricorso a procedimenti automatici, ponendo le questioni dal loro punto di vista, e ponendo l'attenzione sul fatto che l'esperienza viva di ricerca è essenziale per mantenere quanto di positivo c'è nella tradizione metodologica attuale, aliena dai procedimenti automatici.

Questo è il quadro in cui si colloca il Seminario di cui presentiamo le relazioni, che ha cercato di sollecitare un gruppo di studiosi particolarmente qualificato ad esprimere idee e suggerimenti circa la possibilità di integrazione dei diversi procedimenti informatici applicati ai vari settori e momenti della ricerca umanistica, in un ambiente uniforme, che permetta il passaggio di dati da un programma all'altro e da un tipo di ricerca all'altro in modo facile e lineare.

I risultati ci sembrano di grande interesse, e tali da indicare ulteriori sviluppi, di là dalla specificità delle singole soluzioni proposte. Non sarà inutile indicare i temi più importanti emersi dai lavori del Seminario. Il contributo di Luigi Cerofolini[3] traccia un quadro sintetico ma chiaro e preciso dell'ambiente presente e soprattutto futuro nel quale si devono collocare i progetti umanistici. Essi dovranno essere rivolti a studiosi che si suppone dispongano di una stazione di lavoro singola, personale, ma completa delle necessarie «periferiche» per la gestione delle immagini, del suono, e della comunicazione (e un tale tipo di stazione è presupposto dal progetto di Manfred Thaller,[4] rivolto alle ricerche storiche, ma non solo ad esse); ma soprattutto inserita in modo vitale in una rete di calcolatori che costituirà una «biblioteca globale», all'interno della quale lo studioso avrà a disposizione i risultati (ivi compresi i dati e i programmi) delle ricerche di tutti i colleghi. Soprattutto è da rilevare che lavorare sulla base di una rete di comunicazione obbliga in sostanza lo studioso a rendere compatibile il proprio lavoro di ricerca con quello dei colleghi, salvaguardando nel contempo la completa libertà dei contenuti specifici e

3. Docente di matematica applicata presso l'Università degli Studi di Bologna, è prematuramente scomparso il 22 settembre 1992. La dedica di questo volume alla sua memoria vuole essere la testimonianza di quanto la sua opera sia stata apprezzata anche in ambito umanistico.

4. Max-Planck-Institut für Geschichte, Göttingen.

dei risultati scientifici.

Se questo è il quadro riguardante l'ambiente tecnico in cui opera lo studioso, non meno interessante è quello che riguarda l'ambiente metodologico, qui considerato soprattutto nel campo delle basi di dati.[5] Sia Manfred Thaller sia Anne-Marie Guimier-Sorbets[6] trattano della necessità di organizzare e codificare i dati in modo che sia indipendente dalle successive utilizzazioni che gli studiosi vogliano trarne, individualmente e specificamente. Se la Guimier-Sorbets propone dei modi di trasferire dati da un *fichier* all'altro, Thaller giunge a teorizzare un vero e proprio linguaggio (in sostanza un *mark-up language* del tipo di quelli proposti per i testi) con il quale strutturare e descrivere i dati, che peraltro vengono codificati esattamente come si trovano nelle fonti. Si comprende bene come un simile procedimento sia l'unico capace di dare la possibilità di sfruttare i dati per ricerche anche molto diverse da quelle che hanno dato origine alla loro raccolta in una base di dati da parte di un determinato gruppo di ricerca.

Quest'ultima proposta mette anche in rilievo la possibilità di utilizzare programmi del tipo «sistemi esperti» per compiere operazioni «sintetiche» (e non più soltanto analitiche) sui dati, passando, secondo la felice espressione di François Djindjian,[7] «dal trattamento dell'informazione ai processi cognitivi». L'interesse di un tale sviluppo è dato naturalmente non dalla possibilità di far compiere alla macchina un'attività «critica» sui dati, ma di precisare e esplicitare da parte dello studioso le parti logiche e formalizzate della propria metodologia.

Credo che per tutti questi motivi, e per molti altri che il lettore avrà modo di trovare leggendo i contributi che presentiamo, il Seminario, e di conseguenza questo libro, possa essere di non poca utilità, sia a chi si accosti per la prima volta ai problemi in esso trattati, sia per promuovere un dibattito da parte di chi tali problemi veda in modo eventualmente

5. Il campo dell'analisi di testi è da considerasi in questo senso più evoluto, in quanto esistono almeno tre iniziative di vasto respiro dedicate alla standardizzazione dei modi di codificare testi in *machine readable form*: SGML, TEI, SOFABED (Standard Open Formal Architecture for Browsing Electronic Documents, del Davenport group: cf. archivio in «ananonymous ftp, ftp.ora.com», su internet).

6. Centre de recherche sur les Traitements automatisés en archéologie classique, CNRS-Université de Paris X.

7. CNRS - UPR 315.

diverso. Tale dibattito continuerà certamente in seno alla ricerca di cui abbiamo parlato sopra; ed inoltre nel gruppo di discussione elettronica su rete bitnet (e internet) che nel frattempo si è costituito. Per chi fosse interessato, l'indirizzo (elettronico) è: trtidu2 at itcaspur.

Questo libro è dedicato alla memoria di Luigi Cerofolini.

LUIGI CEROFOLINI †

L'AMBIENTE TECNOLOGICO PER L'INTEGRAZIONE

Se si vuole affrontare correttamente il problema della collaborazione tra l'informatica e le discipline umanistiche, ma lo stesso principio vale anche per le altre discipline scientifiche, occorre tener presente che alcuni aspetti propri delle singole discipline che intendono avvalersi di metodi informatici non possono essere immediatamente recepiti dai puri esperti in informatica.

Non è corretto che uno studioso umanista demandi all'informatico compiti e attività propri della sua disciplina, come non lo è per l'ingegnere che si occupa di scienza delle costruzioni, o per il medico che studia le attività del sistema nervoso o di quello circolatorio. È molto più semplice che ciascuno di questi studiosi apprenda l'uso di alcune tecnologie e dei metodi propri dell'informatica per applicarli correttamente al proprio settore di studio.

Si tratta infatti di veri e propri strumenti di lavoro che, come spesso è avvenuto nell'esercizio delle arti, possono essere messi a punto e perfezionati soltanto da chi pratica un'arte specifica, senza dar luogo a un equivoco scambio di responsabilità.

In passato, prima della diffusione dei personal computer, è accaduto che i Centri di calcolo fossero considerati il luogo al quale rivolgersi per implorare che un tecnico scrivesse un programma più o meno rispondente a esigenze che spesso non riuscivano ad essere facilmente comprese.

L'indirizzo che oggi sembra prevalere è quello di usare dei sistemi aperti, ovvero non legati a una particolare macchina, a una ditta costruttrice, né a una particolare cultura informatica. Tra questi, Unix è un sistema operativo che mette a disposizione degli utenti strumenti molto potenti nella forma più completa e più ricca, anche dal punto di vista della presentazione e della documentazione.

In questa sede, ci soffermeremo piuttosto su alcuni aspetti tecnologici, per riuscire a comprendere meglio le potenzialità dello sviluppo tecnologico in atto, al fine di pianificare nel modo migliore le attività di ricerca e di applicazione dei metodi informatici. Ci occuperemo, in particolare,

delle possibilità offerte dalle stazioni di lavoro attualmente disponibili sul mercato e delle prestazioni che presto saranno in grado di assicurare, con costi anche molto contenuti. Tratteremo anche l'argomento dell'elaborazione dei testi, che è al centro dell'interesse di questo Seminario.

L'evento più straordinario del progresso tecnologico degli ultimi anni è certamente rappresentato dall'evoluzione della componentistica, che consente di realizzare componenti elettronici di dimensioni molto ridotte in cui vengono inseriti milioni di transistor.

Questo ha consentito di costruire i moderni calcolatori, che funzionano mediante i cosiddetti componenti di elaborazione, le Central Processing Unit (CPU). Per quanto riguarda le memorie di cui dispongono i calcolatori, si pensi che per costruire una memoria di 4 Mbyte, ovvero da 4 milioni di byte, sono necessari 4 milioni di transistor più quattro milioni di condensatori e qualche centinaio di migliaia di transistor per le interconnessioni e le amplificazioni. I componenti non si usano soltanto per i moduli di elaborazione, ma anche per quelli grafici: infatti, per molti aspetti, oggi la grafica è ritenuta un elemento essenziale dei procedimenti di calcolo.

Un altro punto di grande attualità è quello delle reti di calcolatori, limitandoci qui a trattare dei dispositivi che consentono ai calcolatori di comunicare fra di loro. Le opportunità di cui oggi si dispone consentono di scambiare dati tra due calcolatori con una velocità dell'ordine di milioni di bit al secondo. Se la trasmissione avviene all'interno di reti geografiche, con altre nazioni o continenti, la velocità arriva a decine di milioni di bit al secondo; all'interno delle reti locali – quelle cioè che mettono in comunicazione calcolatori che distano tra loro qualche centinaio di metri, fino a qualche chilometro – si tratta addirittura di centinaia di milioni di bit al secondo. Nel volgere di alcuni anni, si parla di cinque anni, non vi sarà più differenza tra i vari tipi di comunicazione, grazie alla tecnologia delle fibre ottiche, attraverso le quali l'informazione viaggia con la velocità di miliardi di bit al secondo.

Si realizzerà così il villaggio globale, in cui i dati potranno essere condivisi in maniera trasparente con tutto il mondo. Basterà connettersi a questa ideale biblioteca globale, per accedere in maniera immediata, istantanea, ai dati che più ci interessano, superando i problemi derivanti dalle connessioni odierne, più o meno lente.

L'altro fenomeno che sta modificando in maniera radicale il nostro lavoro consiste nell'evoluzione delle periferiche. Per fare soltanto un esempio, si pensi ai miliardi di byte che possono essere registrati su un compact disk in tutto simile a quelli usati per le incisioni musicali. È anche possibile raccogliere diversi dischi ottici in un dispositivo del tipo juke-box, per poterli consultare contemporaneamente. Oltre ai moduli grafici tradizionali, quelli visivi, sono in grande sviluppo anche i settori del suono e dell'immagine animata. A questo proposito, occorre tener presente che, quando si parla di tecnologie grafiche integrate con suoni e immagini, si fa spesso riferimento ai concetti di compressione - poiché la mole di dati di un'immagine video è enorme - e di rete, per la trasmissione: non è pensabile, infatti, che il lavoro prodotto da ognuno si esaurisca in un ambito di utenza ristretto; al contrario, è necessario che confluisca in grandi banche di dati, a disposizione di tutti.

Sebbene già molti calcolatori abbiano le caratteristiche di cui abbiamo fatto cenno, si lavora ancora su macchine con prestazioni inferiori. Molto presto si diffonderanno stazioni di lavoro capaci di elaborare 50 milioni di istruzioni al secondo, dotate di 32 Mbyte di memoria di lavoro, con dischi da 1 Gbyte, ed elevata qualità di elaborazione grafica e del suono.

Un altro fenomeno di estrema importanza è rappresentato dai sistemi distribuiti, dal momento che non si mira più ad una concentrazione di attività. Ai sistemi distribuiti corrispondono, a livello di software, dei protocolli di rete che hanno, tra le caratteristiche fondamentali, l'elaborazione in tempo reale, essenziale soprattutto per la trasmissione delle immagini. Il file system, che consente di accedere in maniera trasparente - a chiunque sia autorizzato - ai dati di interesse comune, dovrà essere un elemento del sistema operativo distribuito sulle macchine di tutti gli utenti che hanno accesso a un grande calcolatore.

Altro aspetto di notevole importanza è il paradigma di client e server: quando si accede a una banca dati o a un programma residenti su un altro calcolatore, magari di un'altra città, si attiva il server, che ha il compito di rendere un servizio all'utente (client), per l'accesso a una macchina remota. La propria stazione di lavoro diventa, in quel momento, un semplice strumento in grado di visualizzare il processo di calcolo che si compie a distanza. Per far sì che questo possa realmente accadere, è necessario raggiungere un alto livello di standardizzazione: nel formato

dei file, nei protocolli di rete, nei sistemi di elaborazione, nei programmi. In questo contesto, occorre citare nuovamente Unix, che, fin dall'inizio, ha avuto il pregio di essere un sistema aperto. È evidente, infatti, che l'integrazione tra sistemi operativi di tipo proprietary è praticamente impossibile, e non è neanche realmente perseguita dalle ditte costruttrici. Soltanto l'esplicita richiesta da parte dell'utente di sistemi aperti può imporre la strategia di standardizzazione e di integrazione degli strumenti e delle idee.

Passando poi ai problemi posti dall'elaborazione di testi, occorre anzitutto precisare che i requisiti fondamentali cui deve rispondere un buon programma per l'elaborazione di testi sono quelli della massima qualità grafica e tipografica. Quando si parla di grafica, si mira ad ottenere un buon risultato visivo sul video e sulla stampante. La tipografia, invece, è estremamente importante ai fini dell'estetica della pagina stampata, dell'interesse e dello stimolo che è capace di suscitare nel lettore, della leggibilità stessa di un testo, con conseguenze rilevanti perfino nel modo di scrivere un testo. Un altro requisito estremamente importante è quello dell'integrazione dei dati, nel senso della loro trasportabilità. Per questo, è molto importante che i file siano «onesti», non abbiano cioè informazioni nascoste «a tradimento» dal sistema, e che quindi possano essere agevolmente trasportati da una macchina a un'altra.

Uno degli aspetti fondamentali dell'integrazione riguarda l'impostazione culturale e tecnologica dei sistemi operativi. Infatti, non è più pensabile di dover trascorrere mesi per imparare ad usare un editor o i comandi di un nuovo sistema operativo, ogniqualvolta si cambi calcolatore, anche perché la tecnologia è in continua evoluzione: è un problema di «disponibilità» del sistema operativo alle necessità dell'utente e ai progressi della tecnologia.

Per quanto riguarda l'architettura dei sistemi di elaborazione, vi sono due impostazioni fondamentali. La prima è quella dei sistemi WYSIWYG (What You See Is What You Get); poiché, in genere, se ne sente parlare con soddisfazione, non starò ad elencarne i pregi, ma mi limiterò a parlare dei difetti. In primo luogo, spesso il video non è in grado di mostrare appieno l'aspetto che avrà la stampa finale; in più, l'utente, che non è tipografo di professione, è indotto a commettere errori talvolta anche grossolani di composizione: il corpo, la giustezza, l'impaginazione. La facilità d'uso, poi, ha spesso un risvolto negativo determinato dalla

scarsità delle scelte e delle possibilità previste dal sistema. D'altra parte, l'interazione con il sistema è tollerabile nel caso di testi brevi, non certo nel caso di testi di centinaia di pagine, che richiederebbero un lavoro ingente per apportare modifiche e correzioni all'impaginato. In aggiunta, i dati diventano inutilizzabili da altri sistemi, perché non sono più «onesti».

I sistemi non interattivi sono invece corredati da un'ampia serie di regole tipografiche, che vengono associate dal sistema alle indicazioni fornite dall'utente con comandi descrittivi del tipo: questo è un titolo, qui inizia un nuovo paragrafo, e così via. La caratteristica più importante dei sistemi non interattivi è, tuttavia, l'assoluta «onestà» dei file che vengono prodotti. Altri aspetti degni di nota sono la flessibilità e la programmabilità del sistema, che non risulta in alcun modo vincolato a opzioni predisposte. Il sistema che risponde a questa logica è Unix, che è in grado di soddisfare anche i problemi di grafica e di networking.

In conclusione, se si guarda ai problemi dell'integrazione, occorre avere costantemente presente un'idea che consenta di rimanere sempre responsabili delle scelte e degli strumenti per l'elaborazione, anche a costo di dover sacrificare alcune opportunità già predisposte; quest'idea è quella dei sistemi aperti.

WILHELM OTT *

EDIZIONE CRITICA E GESTIONE DI TESTI
IL PACCHETTO TUSTEP

Venti anni fa, in un convegno come questo, lo scopo principale di una relazione sul tema «edizione critica e gestione di testi» sarebbe stato quello di convincere i colleghi che col calcolatore era finalmente disponibile un nuovo strumento molto efficace da utilizzare anche nelle discipline umanistiche. Anche nel nostro settore di studi non sarebbe più stato possibile fare ricerca senza usare i metodi dell'informatica.

Oggi, quasi ogni studente ha a disposizione un calcolatore paragonabile in capacità a quelli di cui disponevano i centri universitari di calcolo degli anni Settanta, e può utilizzare anche pacchetti di programmi allora neppure immaginabili.

Il Word Processor non ha soltanto sostituito la macchina da scrivere, eliminando al curatore di un'edizione la fatica di scrivere di nuovo un testo, dopo ampie correzioni e revisioni. L'editoria stessa ha fatto la sua comparsa sulla scrivania dello studioso, che può così realizzare, mediante le tecniche del *Desk Top Publishing*, la composizione e l'impaginazione della sua edizione critica, invece di consegnare un testo scritto a macchina alla casa editrice che provvede alla stampa e alla pubblicazione.

Anche per la gestione di basi di dati è disponibile una varietà di sistemi per il PC, che fornisce un'insolita flessibilità al tradizionale archivio cartaceo, lo strumento tipico dello studioso di discipline umanistiche.

A prima vista, dandosi questo scenario, sembra che ci resti un solo problema: l'integrazione dei diversi procedimenti offerti dagli strumenti per la elaborazione di testi, per la gestione di dati e della preparazione per la stampa. Questo, però, è uno degli equivoci più comuni.

La preparazione di edizioni critiche è uno dei settori nei quali si cercava da tempo di servirsi di strumenti tecnici. In un articolo del 1965 dal

* Tübinger Zentrum für Datenverarbeitung.

titolo *Die Technifizierung der Edition: Möglichkeiten und Grenzen*[1] («La meccanizzazione dell'edizione: possibilità e limiti»), Helmut Praschek distingue sette fasi di lavoro: la terza fase (la collazione), la quarta fase (la valutazione dei risultati della collazione allo scopo di caratterizzare i testimoni e di investigare sulla loro interdipendenza) e la quinta fase (la costituzione del testo dell'edizione) occupano la maggior parte dell'articolo. Per nessuna di quelle 3 fasi, che erano centrali per Praschek, e lo sono per il lavoro scientifico di preparazione di un'edizione critica, gli strumenti menzionati sopra (WP, DTP, base di dati) possono offrire un aiuto significativo. Sono necessari strumenti diversi da quelli preparati per l'automazione dei lavori in ufficio.

Fornire tali strumenti, adatti al trattamento scientifico di testi, è lo scopo principale del sistema TUSTEP. Oltre a contenere funzioni che non si trovano nei più diffusi sistemi per l'elaborazione di testi, gli strumenti che compongono il sistema TUSTEP coprono tutte le fasi del lavoro di un progetto di ricerca umanistica: la registrazione dei dati, le varie fasi di analisi, di elaborazione scientifica, di documentazione, fino alla stampa con qualità tipografica professionale, e fino alla loro organizzazione in una base di dati per la ricerca interattiva. Il tema generale di questo convegno, l'integrazione dei diversi strumenti di lavoro scientifico su dati testuali, è dunque uno dei punti centrali di TUSTEP.

Vorrei dividere la mia relazione in tre parti:

- le funzioni di base offerte dal TUSTEP per il trattamento scientifico di testi,
- l'applicazione degli strumenti relativi alla preparazione di edizioni critiche e alla gestione dei testi,
- l'integrazione.

Ma 30 minuti sono troppo pochi per seguire questo schema. Preferisco quindi concentrarmi sugli elementi fondamentali dell'architettura, della «filosofia» del sistema. Per una descrizione più completa posso rimandare, fra l'altro, all'articolo «Il sistema TUSTEP nell'edizione critica di

1. In: H. Kreuzer und G. Gunzenhäuser (eds.), *Mathematik und Dichtung. Versuche zur Frage einer exakten Literaturwissenschaft*, München, Nymphenburger Verl., 1965, pp. 123-142.

testi», nel volume *Trattamento, Edizione e Stampa di Testi con il Calcolatore*, pubblicato nel 1989 a cura di Giovanni Adamo (è il testo di una relazione tenuta nel 1986, a Roma, durante una giornata di studio organizzata dal Prof. Orlandi).

TUSTEP è un acronimo dell'espressione tedesca «Tübinger System von TExtverarbeitungs-Programmen», cioè, «sistema di programmi per l'elaborazione testi»; in inglese, «TUebingen System of TExt Processing programs». La denominazione è del 1978, ma il sistema cominciò ad essere sviluppato fin dal 1966.

Vorrei mettere in evidenza tre aspetti presenti nella denominazione del sistema.

Primo: il plurale «programmi». Non si tratta di una *black box*, un super-programma, ma di un pacchetto di programmi relativamente autonomi. Ciascuno dei programmi è limitato più o meno a compiere una sola delle funzioni di base del trattamento scientifico di testi.

Secondo: la parola *Verarbeitung*, che traduco con *trattamento*. Ho scelto questo termine in un'epoca in cui non vi erano influenze dell'espressione inglese *Word Processing*, che si riferisce sostanzialmente alle fasi di registrazione, correzione e formattazione di testi. Se ho capito bene, anche l'espressione italiana «elaborazione di testi» è limitata alla manipolazione del «testo contenuto in un *file* per dare un aspetto piacevole e comodo da leggere al testo stesso, quando questo viene stampato» (Mondadori, *Dizionario di Informatica*, 1990, s.v. *Word Processing*). Nel contesto di TUSTEP, se parliamo di elaborazione di testi, i *testi* vengono considerati dai singoli programmi come *dati*, e l'elaborazione consiste in funzioni di base (analisi, selezione, trasformazione, ordinamento di dati testuali) previste dai singoli programmi.

Terzo: questa collezione di programmi diventa un sistema grazie all'integrazione in un ambiente uniforme che permette di combinare a piacere le varie funzioni, secondo le esigenze del problema da risolvere e secondo le esigenze di un progetto che, dalla registrazione dei dati fino alla pubblicazione dei risultati, prevede molte fasi di lavoro con relativi problemi. Questa integrazione è una delle condizioni principali per il lavoro giornaliero, perché essa sola evita il bisogno di trasformare i dati in base ai diversi lavori che si possono rendere necessari durante la ricerca.

Nel 1976, all'occasione di un congresso dell'*Association for Literary*

and Linguistic Computing[2] ho descritto la tecnica che permette questa integrazione come segue: «This basic idea is as simple as it is fundamental to a flexible text processing system: the output of any one program, including the composing program, may serve as input for any one other program» (p. 32); «what counts here is that there is one system of programs covering the text processing requirements of an editor from the first input up to the final output» (p. 35).

Nonostante questa caratteristica, per i casi nei quali si rendono necessari strumenti supplementari (per esempio di analisi statistica) o nei quali altri strumenti risultino essere più adattati a un certo scopo o disponibili più facilmente, TUSTEP facilita anche l'integrazione di tali strumenti. Infatti contiene un programma di trasformazione dei dati dal formato TUSTEP nel formato usato dal sistema operativo e viceversa (per esempio, su sistemi MS-DOS o Unix, i file dal formato TUSTEP si trasformano in file formato ASCII e viceversa; per sistemi operativi della IBM o della SIEMENS, si tratta di un altro formato e della codifica in EBCDIC). Così i risultati ottenuti mediante i programmi TUSTEP possono essere esportati per essere sottoposti ad analisi statistica mediante pacchetti come SPSS, ed i risultati di tale analisi possono essere reimportati in TUSTEP. Con lo stesso procedimento, anche per la registrazione dei dati, altri sistemi possono essere integrati nel procedimento, se per un dato progetto ciò possa risultare più agevole.

Se, per esempio, la segreteria è solita usare il pacchetto WordPerfect o altro per il lavoro d'ufficio, può senz'altro continuare ad usare lo strumento a cui è abituata anche per trascrivere le fonti per una edizione. Lo studioso poi importa i *file*, trasforma la codifica, e continua a lavorare con TUSTEP.

C'è un altro aspetto importante dell'integrazione dei mezzi informatici in un ambiente uniforme, cioè l'indipendenza dal hardware e dal sistema operativo. Né il ritmo dell'installazione di nuovo hardware e software nei centri di calcolo, né il ritmo del progresso tecnico nei minicomputer e workstations sono normalmente sincronizzati con i tempi del lavoro in un progetto di edizione. Il lavoro potrebbe essere ostacolato sensibilmente dall'uso di strumenti informatici se questi non fossero più

2. Vedi «A Text Processing System for the Preparation of Critical Editions», CHum 13 (1979) 29-35.

disponibili da un giorno all'altro a causa di «sviluppo tecnico», o quando ogni quattro anni richiedono un considerevole lavoro di adattamento.

Anche il PC su cui si è iniziato un lavoro può risultare, con il tempo, troppo limitato in capacità, se i dati o i problemi raggiungono dimensioni non previste. Solo se lo strumento utilizzato è indipendente da un certo tipo di calcolatore, si può continuare il lavoro senza interruzione metodologica con un sistema più capace.

Questa indipendenza è una delle proprietà importanti di TUSTEP: il pacchetto contiene non solo i programmi che offrono funzioni di elaborazione scientifica, ma anche tutti gli strumenti necessari per la gestione dei dati e dei programmi, compresi i comandi per creare, assegnare, rilasciare, salvare i file, gli strumenti necessari per definire nuovi comandi, etc. Abbiamo integrato nel sistema tutto questo complesso di comandi di controllo che normalmente sono offerti dal sistema operativo del computer. Così il sistema TUSTEP si presenta all'utente sempre nella stessa veste, e in maniera del tutto autonoma rispetto al computer e al sistema operativo su cui è installato. Infatti, i comandi di logon e logoff (che non esistono sul PC) sono normalmente i soli comandi del linguaggio di controllo del computer che l'utente di TUSTEP debba considerare. Questo è molto importante perché è la garanzia della trasportabilità dei dati e delle procedure da un computer all'altro, per esempio dal PC a un grosso computer da centro di calcolo o viceversa, o da un mainframe sotto MVS a un altro sotto Unix o VMS.

La portabilità è importante non solo per progetti di lunga durata, ma anche per progetti ai quali collaborano più istituzioni scientifiche. Solo se il pacchetto può essere installato su diversi tipi di computer, la collaborazione funziona anche se il hardware disponibile è incompatibile.

Veniamo ora a trattare le funzioni offerte dai singoli programmi di TUSTEP. Per ragioni di tempo, mi limiterò alle funzioni più importanti nel contesto dell'edizione critica.

Una delle funzioni di base è la preparazione del testo e degli apparati critici per la stampa. Siccome TUSTEP vuole essere uno strumento per lavoro professionale, contiene, oltre ad un programma di formattazione per stampanti on-line, anche un programma per la fotocomposizione. Non credo sia necessario ripetere quello che ho già sottolineato molte volte, anche qui a Roma, sulla necessità di avere disponibile uno strumento automatico, che eviti il rischio di errori tipografici, per

preparare la stampa dei dati con qualità tipografica professionale.

Nel contesto di questa relazione credo sia importante considerare un solo aspetto del programma di composizione che lo distingue di altri programmi simili.

L'impaginazione automatica ha due aspetti: uno di tipo economico, che consiste nella soluzione algoritmica di problemi anche difficili dal punto di vista tipografico per la composizione di pagine con apparati diversi a piè di pagina. Questo aspetto è importante non solo per la casa editrice; gli stessi curatori delle opere complete di Leibniz ne fanno uso per la preparazione di un volume all'anno di una «edizione preliminare», «Vorausedition ad usum collegialem», che non potrebbe essere finanziata se preparata con metodi tradizionali.

Il secondo aspetto riguarda l'integrazione del programma di composizione nel sistema.

La composizione e l'impaginazione aggiungono nuove informazioni a un testo, o meglio alle parti fisiche che costituiscono il testo, l'informazione sulla loro localizzazione nel libro (cioè il numero pagina) e all'interno della pagina (il numero della riga). E' necessario disporre di questa informazione per i vari sistemi di riferimento usati, per es. nell'indice delle materie, negli indici di persone, nomi, luoghi, nelle concordanze, per riferimenti su altri passi del testo, etc.

Il programma di composizione di TUSTEP rende disponibile questa informazione mediante una delle caratteristiche del formato interno dei file di TUSTEP. Come in un libro stampato, anche in TUSTEP ogni testo si presenta con una struttura a tre dimensioni. Come un libro è costituito da singole righe (prima dimensione) che costituiscono le pagine (seconda dimensione) che formano il libro, così anche in un file di TUSTEP ogni testo è soddiviso in «records» o blocchi di dati, unità di registrazione che al momento della composizione corrispondono a una riga di testo nel libro stampato. Questa struttura di dati non si trova normalmente nei file di molti sistemi operativi, mentre è comune in altri. In TUSTEP, ogni record ha un indirizzo numerico univoco che non viene cambiato se non con comando. Esso consiste nel numero di pagina e nel numero di riga. Benché si possa utilizzare questa numerazione per altri scopi (p.e. nel caso di una bibliografia, avrebbe senso usare la parte che corrisponde al numero di pagina per numerare le singole unità bibliografiche), nel contesto della composizione (o, più generalmente, nel contesto di testi

correnti) è opportuno usare questo numero di record per indicare il numero di pagina e di riga del testo stampato. Nella registrazione di dati, si potrebbe usare la paginazione e la divisione in righe del libro o del manoscritto da cui si trascrive il testo. Se verso la fine del lavoro editoriale il testo viene composto e impaginato dal programma di composizione, il numero dei singoli records di un text file rappresenta la nuova divisione del testo in pagine e righe fatta dal programma. Mentre altri sistemi di composizione producono solo un file che controlla l'unità di fotocomposizione, il programma di composizione di TUSTEP prepara un secondo file che è identico al file di entrata per quanto riguarda la codifica, ma se ne discosta in quanto mostra la nuova divisione in records e quindi la numerazione di questi records secondo la divisione in righe e pagine fatta per la macchina compositrice. Così, tutte le informazioni di posizione nel libro che ne risulta sono disponibili per l'analisi ulteriore del testo, per la preparazione dell'indice, delle concordanze, etc.

Questa struttura dati è usata dal TUSTEP per i cosiddetti «text file». Essa riflette la struttura più o meno fisica di un testo scritto o stampato. Va da sé che altre strutture possono essere rappresentate, o mediante la divisione in records (la lunghezza di un record è limitata a 32.000 caratteri), o mediante una codifica adeguata. Così, la divisione in capitoli e paragrafi, o, per la poesia, in poemi, strofe e versi, o, per il dramma, in atti e scene non si fa normalmente mediante il numero dei record, ma mediante una codifica che indica l'inizio o il cambio e l'identificazione di una tale unità logica. Questa struttura può essere di una complessità arbitraria, secondo le necessità di ogni singolo progetto.

Mediante questa struttura, si ha la possibilità di passare da una registrazione di testi correnti ad un'altra di dati o informazioni strutturati secondo una classica base di dati. Così, quando si lavora con TUSTEP, anche la necessità di usare un data base system è normalmente abbastanza remota. Nel'Istituto della Storia della Medicina dell'Università di Tübingen, una bibliografia di ca. 100.000 titoli è stata memorizzata e viene interrogata mediante TUSTEP. (Naturalmente anche TUSTEP ha i suoi limiti, ma non sono molto rilevanti. Per esempio, l'estensione di un singolo file che può essere interrogato o corretto mediante l'editor è limitata a 7 GB, equivalenti a più di 1000 edizioni complete della Bibbia in lingua latina).

Torniamo alla struttura di un text file e alla gestione di testi con

TUSTEP. Tale struttura è fondamentale per TUSTEP, perché è utilizzata da molte funzioni di base per le quali un riferimento univoco è necessario. Una di queste funzioni è la collazione di molteplici versioni di un testo. Il programma di collazione del TUSTEP permette di collazionare due versioni alla volta; ogni valutazione dei risultati deve essere fatta con altri programmi.

Quando solo due versioni devono essere collazionate, basta come risultato un protocollo nel quale il testo delle due versioni viene stampato, l'uno sotto l'altro, segnalando le differenze. (Anche sul programma di collazione non posso parlare in detaglio; chi si interessa, può leggere il paragrafo rispettivo dell'articolo già citato, o uno degli articoli[3] che ho pubblicato sulla collazione automatica). Già per lo scopo di questo protocollo, i numeri di pagina e di riga sono utili per la localizzazione delle differenze nel testo, perché normalmente non tutto il testo viene stampato ma solo le righe dove si trovano differenze o varianti.

I riferimenti diventano però indispensabili se più di due versioni di un testo devono essere collazionate. Come ho già detto, TUSTEP collaziona solo due versioni alla volta, ma prevede la possibilità di accumulare i risultati della collazione successiva di tutte le versioni di un medesimo testo. I risultati della collazione, cioè le varianti, non vengono stampati, ma ricordati in un file. In questo file devono essere contenute tutte le informazioni necessarie per identificare univocamente una variante. Queste informazioni consistono in un riferimento univoco alla versione di base (numero pagina, numero riga e numero parola nella riga), seguito da un codice per identificare la versione del testo che contiene questa variante, poi da un codice che indica il tipo di variante (omissione, sostituzione, aggiunta), e finalmente dalla parola o delle parole varianti.

Questo è tutto ciò che il programma di collazione offre (ho omesso

3. «The Output of Collation Programs», in: D. E. Ager *et al.* (eds.), *Advances in Computer-aided Literary and Linguistic Research*, Birmingham, The University of Aston, Dept. of Modern Languages, 1979, pp. 41-51; «Transcription errors, variant readings, scholarly emendations: software tools to master them», in: Association Internationale Bible et Informatique, *Actes du Second Colloque International «Bible et Informatique: méthodes, outils, résultats»*, Jerusalem, 9-13 juin 1988, Paris-Genève, Champion-Slatkine 1989, pp. 419-434; «Mehr als Kollationshilfe: Automatischer Vergleich als Editionswerkzeug«, in: *Mathesis rationis. Festschrift für Heinrich Schepers*, Münster, Nodus 1990, pp. 349-372.

qualche altra informazione che può essere inserita in questi records, come il testo originale della versione di base, o l'informazione di posizione della variante nel testo collazionato). Per ogni tipo di valutazione, ci si può avvalere di uno degli altri programmi di TUSTEP. Per ottenere, per es., una stampa a partitura, prima si deve fare un ordinamento delle varianti accumulate nel file di output del programma VERGLEICHE. In questo ordinamento, il primo criterio è costituito dal luogo della variante (numero di pagina, riga, parola), il secondo criterio dalla codifica usata per identificare la versione che contiene la variante. Per ottenere invece una lista che corrisponde a un apparato critico, l'ordinamento deve avere come secondo criterio, dopo il luogo di variante, la stessa parola variante, affinché varianti identiche siano raggruppate insieme. La codifica della versione sarà il terzo criterio di ordinamento.

Per studiare la genealogia con metodi statistici, si parte da questa lista, se ne estraggono i gruppi di manoscritti che presentano la stessa variante nei singoli passi, e se ne fa un ordinamento alfabetico. Dopo il sort, viene calcolata la frequenza dei singoli gruppi, e il file che ne deriva può essere ordinato secondo l'ordine della frequenza dei singoli gruppi, o si può esportare questo file per farne una analisi statistica con gli appositi pacchetti di programmi statistici.

Abbiamo visto come il numero di record che fa parte della struttura di un text file di TUSTEP, è usato come parte essenziale dell'identificazione del luogo della variante quando questa viene registrata nel file. Con tale metodo le procedure menzionate di accumulazione, ordinamento, stampa a partitura, preparazione di apparati etc. non si limitano ai casi nei quali il testo completo di tutti le versioni prese in considerazione è stato registrato mediante tastiera o lettore ottico. Questo metodo permette di integrare anche i risultati di una collazione fatta con mezzi tradizionali, con un foglio di carta su cui è stampato, in prima riga, il testo di base, e su cui si registrano nelle altre righe le varianti trovate nelle versioni prese in esame. Se si sono preparati questi fogli di collazione in maniera che la prima riga contenga il numero pagina e il numero riga, è facile registrare le varianti nella stessa maniera richiesta dal TUSTEP, aggiungendo il numero di parola (contando le parole) nella riga e la codifica stabilita per l'inserimento o la sostituzione, o l'omissione nella parte antistante il testo della variante stessa. Così è possibile non solo inserire questi dati negli altri dati trovati per mezzo della collazione automatica,

ma è persino possibile ricostruire automaticamente il testo completo della versione collazionata a mano, prendendo il file che contiene il testo di base (stampato nella prima riga del foglio di collazione) e il file che contiene la registrazione delle varianti come file di input per il programma di correzione automatica.

Questo programma interpreta la registrazione delle varianti non come descrizione di uno stato esistente, ma come istruzioni di correzione. Presupposto che la collazione a mano sia stata eseguita con la completezza e l'esattezza richesta, da tale procedura risulta il contenuto esatto del testo della versione collazionata, con la sola eccezione della sua divisione in pagine e righe.

Come ho detto, posso soffermarmi soltanto su alcuni punti centrali. Devo omettere tutto quello che la generazione automatica di indici, di liste di frequenze, di liste di forme (anche inverse), di concordanze può significare in rapporto alla preparazione di un'edizione. Debbo omettere anche le possibilità offerte da TUSTEP per trasformare un testo non mediante operazioni interattive con l'editor, ma a mezzo di un programma che permette di formulare delle regole per i cambiamenti previsti, per es. per la normalizzazione dell'ortografia di un testo. TUSTEP offre un ambiente di programmazione a livello molto alto per questo scopo; inoltre, i risultati di tali programmi possono essere controllati in una maniera ottima mediante la collazione automatica delle due versioni di entrata e di uscita di una tale procedura: cosi, non solo le regole (che esistono nella forma commandi e parametri per il TUSTEP) con cui si tratta un testo sono ben definite e documentate, ma anche il loro effetto può essere controllato facilmente, grazie all'integrazione di tutti gli strumenti in un sistema uniforme.

Concludendo vorrei tornare a sottolineare il fatto che i singoli comandi offrono solo delle funzioni fondamentali di trattamento del testo. Il TUSTEP non offre nessuna soluzione per nessun problema. Questa restrizione sembra qualche volta un po' esagerata, se, per esempio, si riguarda che TUSTEP non contiene neanche un commando per preparare delle liste alfabetiche di forme che si trovano in un testo. Invece, si è costretti a combinare i programmi di decomposizione testo, di aggiunta di chiavi d'ordinamento, dell'ordinamento stesso (SORT), di preparazione di articoli dell'index dopo il sort che preserva una sola di molte entrate successive identiche e ne aggiunge la frequenza e i

riferimenti, e finalmente il programma di formattazione per la stampa. Per chi vuole avere un comando INDEX, TUSTEP offre i mezzi per costruire un tale comando che invoca i singoli programmi necessari con una sola chiamata; possono essere previsti, in un tale comando, parametri, per esempio, per la lingua del testo. Questi parametri possono essere tradotti nei parametri interni necessari per procurare l'ordine alfabetico richiesto per la lingua del testo. Ma è l'utente che deve fare questo lavoro. TUSTEP intenzionalmente, quindi, non vuol essere un sistema esperto, neanche per problemi semplici come l'ordine alfabetico necessario per una lingua come il greco o lo spagnolo. Il sapere necessario per risolvere un certo problema deve essere fornito dall'utente che cosi è costretto a scomporlo nei suoi elementi fondamentali. Secondo me, questa è l'unica possibilità che permette (e costringe) l'utente del computer ad assumere la responsabilità completa dei risultati, perché sa quali sono le funzioni elementari che ha combinato per trovare la soluzione di un problema, e perché ha la possibilità di controllare perfettamente gli effetti di tutti i passi eseguiti. Quelli, quindi, che potrebbero apparire i limiti di TUSTEP sono invece le colonne d'ercole necessarie perché il mestiere di critico e di filologo trovi nell'informatica non un facile e suggestivo alibi per le proprie responsabilità ma uno strumento che valorizza le caratteristiche fondamentali del suo lavoro scientifico.

FRANÇOIS DJINDJIAN[1]

L'ARCHÉOLOGIE COGNITIVE UNE RÉPONSE AU PROBLÈME DE L'INTÉGRATION DES TECHNOLOGIES DE L'INFORMATION EN ARCHÉOLOGIE

Depuis son institutionnalisation à la fin du 18ème siècle, l'archéologie ressent le besoin d'un statut scientifique, qu'elle recherche dans les tentatives de constitution d'une «épistémologie pratique», besoin qu'elle partage avec d'autres disciplines des sciences humaines, comme l'histoire, l'ethnologie ou la sociologie.

Une des raisons majeures des difficultés rencontrées est que l'archéologie n'est pas unique, elle est multiple. Depuis les débuts du 19ème siècle, sont en effet apparues des archéologies parallèles, complémentaires et contradictoires, ambitieuses ou limitées, toujours différentes dans leurs moyens et leurs objectifs.

Une autre raison à ces difficultés est que la technologie contemporaine, par le niveau exigé pour la maitriser, a beaucoup obscurci, ces trente dernières années, le discours épistémologique, abandonnant l'archéologue désemparé entre un acte de foi scientiste, un abandon littéraire ou idéologique, ou un renoncement dans une frustration documentaire.

1. L'archéologie, une science ?

L'archéologie fait partie des «sciences humaines», disciplines dont les essais de formalisation depuis près d'un siècle font l'objet de travaux incessants.

De toutes les sciences humaines, l'archéologie est certainement une des plus complexes à formaliser : son domaine est le plus vaste par définition, puisqu'il couvre tous les sujets qui constituent les éléments de

* CNRS - UPR 315.

reconstruction des civilisations, et ses moyens sont les plus limités, n'ayant à sa disposition que des informations partielles, biaisées, ou dont la signification a été perdue.

L'archéologie est une science dont les formalismes ne sont que le résultat d'acculturations successives de sciences connexes, avec leurs apports et leurs limites. On peut ainsi affirmer qu'il n'y a pas une archéologie, mais des archéologies dont les prémisses et les conclusions sont différentes.

Les sciences des Antiquités ont apporté la pratique des collections d'objets, la muséologie et les corpus documentaires. Les sciences naturelles ont apporté la chrono-stratigraphie, la théorie de l'évolution, la taxonomie (les typologies) et finalement les «cultures». Les sciences sociales ont apporté les déterminismes dans le comportement des groupes d'individus, sur le plan géographique, écologique, démographique et social. Les sciences physiques ont apporté les notions de mesure, de répétitivité, et de validation des résultats. Les sciences du traitement de l'information, de la sémiologie et des disciplines qui s'en recommandent de près ou de loin, ont apporté un début de formalisation qui a pris souvent des voies technologiques : documentation automatique, analyse des données, systèmes experts, simulations, etc., pour parvenir enfin aux questions méthodologiques de façon explicite et non plus implicite à la différence des archéologies précédentes.

La reconstitution archéologique est un discours dans lequel l'archéologue va se projeter au niveau individuel («filtres culturels individuels») comme au niveau social («modèles, déterminismes à priori»). Une position extrême (A) est de considérer que l'archéologie n'est que cela, déniant par là même à cette discipline tout statut scientifique. A l'opposé, l'autre position extrême (B) est de considérer que l'archéologie peut et doit atteindre un statut de sciences exactes, au mème titre que les autres sciences, qui ont atteintes ce statut. Une position tierce (C) est de considérer que l'archéologie doit définir sa propre épistémologie et son propre statut de sciences humaines sans référence obligatoire et sans complexes vis-à-vis des deux précédents points de vue.

Le premier point de vue A considère que les observables (les données) ne sont pas indépendants de l'observateur et donc des théories. Le second point de vue B admet cette neutralité des données. Le troisième C admet la relation entre faits et théories et entre théories et faits, et recherche les

méthodes permettant de les relier de façon univoque, en évitant les pièges à la fois de la circularité des raisonnements et de la multi-interprétation des résultats.

Le lecteur aura compris que c'est dans le cadre de ce troisième point de vue que nous allons nous placer, pour discuter de l'intégration des technologies contemporaines en archéologie.

2. *Une carricature des approches archéologiques actuelles*

Il est naturellement difficile de vouloir résumer les approches méthodologiques de l'archéologie à quelques schémas simples. On nous le pardonnera, sachant que le but ici n'est pas d'être exhaustif, mais d'introduire, par quelques exemples caractéristiques, le développement qui suivra.

La première approche, désignée généralement par approche empirico-inductive, peut s'exprimer par le schéma suivant :

$$\text{Données} \xrightarrow{1} \text{Structures} \xrightarrow{2} \text{Cultures}$$

La faille de cette approche est que toute structure, par construction, peut devenir culture, car assez rapidement, le raisonnement 2 devient interprétation puis tautologie. C'est l'approche classique des sciences naturelles.

La seconde approche, désignée par approche hypothético-déductive, peut s'exprimer par le schéma suivant :

$$\text{Données} \xrightarrow{1} \text{Modèles} \xrightarrow{2} \text{Cultures}$$

La faille de cette approche est que généralement en archéologie, tous les modèles marchent du fait de l'insuffisance des données et du contenu carricatural des modèles. C'est l'approche classique de la «New Archaeology».

Si tous les modèles marchent et si toutes les structures sont cultures, alors à la fois les partisans des points de vue A et B sont comblés parce qu'ils ont prouvé, les premiers, l'échec du mécanisme 2, et les deuxièmes le succès du mécanisme 1 ... Seuls les partisans du point de vue C sont frustrés. Comment donc sortir de cette frustration ?

3. Du recueil de l'information aux processus analogiques

L'archéologie ne manipule que peu d'informations, au sens strict du terme : elle manipule surtout des objets ou des systèmes d'objets.

L'information, qu'elle s'appelle *données, description, Po ou information intrinsèque* suivant les approches formelles pratiquées, résulte de l'interaction entre les objets et l'archéologue (sa culture et celle de sa communauté scientifique ou de sa société et de ses modèles), guidée par une problèmatique archéologique.

En conséquence, l'information, délivrée par les objets, n'est pas neutre, elle est dépendante (notion de dépendance) : elle n'est pas unique, elle est multiple (notion de guidage). Ces caractéristiques expliquent les limitations cognitives des banques de données archéologiques.

L'information est le résultat d'un *processus analogique*, interaction entre l'archéologue et l'objet ou le système d'objets. En corollaire, l'information n'est pas globale ou universelle, elle est locale, délimitée par le champ d'applications du processus analogique (notion de pouvoir local d'une analogie).

La question fondamentale pour la résolution d'une problématique archéologique est la capacité de l'archéologue à formaliser explicitement un processus analogique, à tous les niveaux de la construction, -étic ou -émic, pour pouvoir en reconnaitre le pouvoir et les limites informationnelles. Citons, par exemple :

- les analogies morpho-techniques et fonctionnelles pour les typologies,
- les analogies ethnologiques, par les modèles de peuplement,
- les analogies sociologiques, pour les déterminismes,
- enfin, les analogies positivistes pour les archéologies elles-mêmes.

En conséquence, un des problèmes majeurs de l'archéologie et sur lequel nous reviendrons plus loin, est l'éventuelle insuffisance sémantique de l'information vis-à-vis des exigences de la construction, et qui, quelque soit le moteur du processus cognitif, est la cause de la faiblesse de la construction, entraînant l'échec, soit par arrêt prématuré, soit par erreur, du à des phénomènes de multi-interprétation.

4. Du recueil de l'information aux processus relationnels

En archéologie, la prééminence historique apportée aux objes ou aux systèmes d'objets a longtemps fait négliger l'intérêt du concept de relations entre objets, à une seule exception, celle des ensembles clos.

Les raisons en sont multiples. La première raison est qu'une relation n'est ni manipulable, ni photographiable, ni mesurable comme un vestige matériel. Une deuxième raison est qu'il y a eu souvent amalgame entre relation et information : par exemple, «être du même type que» est une relation, être l'outil «n. i» est une information. Amalgame puis abus du langage ! La troisième raison est qu'une relation plus encore qu'une donnée nécessite le formalisme d'un processus : nous l'appellerons ici *le processus relationnel.*

On aura naturellement fait le rapprochement entre différents concepts rencontrés dans des disciplines qui peu ou prou concernent l'archéologie : le modèle entités-relations en informatique, les bases de faits et les bases de règles dans les systèmes experts, tableaux de description et tableaux de contingence en analyse des données, etc...

5. Du traitement de l'information aux processus cognitifs

De toutes les expériences logico-mathématiques qu'a vécues l'archéologie ces trente dernières années, en passant par les statistiques, l'intelligence artificielle, l'analyse des données, les systèmes experts, etc..., un des résultats les plus importants, sinon le plus important, est qu'en archéologie, ainsi que vraisemblablement en sciences humaines, les processus cognitifs sont des *processus d'apprentissage*, mettant en oeuvre des mécanismes dynamiques et non pas seulement statiques. Ainsi vaut-il mieux parler non pas de conclusions ou de résultats d'une construction, mais plutôt d'un état cognitif du processus d'apprentissage, dont les modifications successives sont le résultat d'actions de validation, qui, basées sur des fouilles archéologiques, concrétisent un discours archéologique de bout en bout, avec ses nombreuses rétro-actions.

Les processus cognitifs ont recours globalement à des machines logiques (des «moteurs») dont les règles sont les classiques mécanismes inductifs et hypothético-déductifs. Il est une illusion trop répandue de croire que ces règles logiques s'appliquent aussi bien en sciences

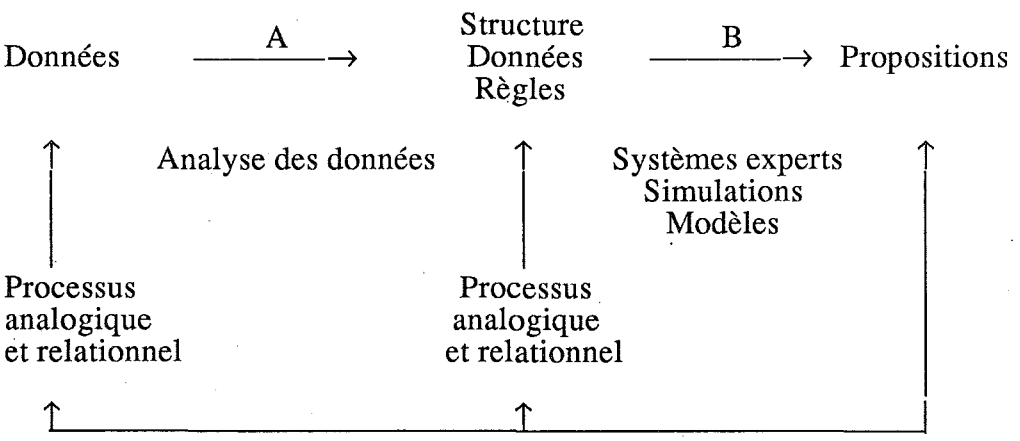
humaines que dans des sciences plus exactes. En effet, la différence entre les données initiales sur lesquelles s'appliquent le mécanisme inductif et les hypothèses sur lesquelles s'appliquent le mécanisme hypothético-déductif n'est pas aussi tranchée du fait même de l'existence du processus analogique précédemment décrit. Les règles logiques elles-mêmes apparaissent comme étant symmétriques, pouvant fonctionner dans la même construction dans un sens (inductif) comme dans l'autre (hypothético-déductif), alors que dans leur définition dans les sciences exactes, elles ne sont pas symmétriques.

C'est bien en fait l'évidence que, en archéologie, les mécanismes logiques sont différents. L'approfondissement de la compréhension de ces mécanismes fera indiscutablement franchir une étape significative pour leur intégration dans une épistémologie pratique.

Les moteurs cognitifs utilisés en archéologie assez globalement se raftachent à deux grandes familles de mécanismes :

- des mécanismes de corrélation ou d'association, générateurs de structures, comme on en retrouve généralement dans toutes les techniques quantitatives qui appartiennent aujourd'hui aux méthodes de l'analyse des données.
- des mécanismes de déduction comme on en retrouve tout aussi bien dans les moteurs d'inférences des systèmes experts, que dans les langages ou modèles de simulation des plus anciens aux plus récents.

Ces mécanismes sont plus complémentaires que concurrentiels ainsi que le montrent des exemples récents où analyses des données et systèmes experts font bon ménage, et comme le figure le schéma suivant :



6. Deux modèles d'épistémologie pratique en archéologie

Le premier modèle inspiré de l'intelligence artificielle, est le modèle logiciste de J. Cl. Gardin (1980, 1991) basé sur l'enchaînement logique de propositions :

$$Po \rightarrow P1 \rightarrow Pi \rightarrow Pi+1 \rightarrow Pn$$

partant de données initiales Po, aboutissant par des mécanismes inductifs ou hypothético-déductifs, à une proposition finale, censée être d'un ordre explicatif plus élevé. Le mécanisme cognitif utilise les principes utilisés dans les systèmes experts, de base de faits et de bases de règles, avec un mécanisme de production du type :

Si p dans la base de fait et si p → q dans la base de règles
alors q dans la base de fait.

Les contextes C1, C2 ... Cn définissent les domaines de validité de chacune des règles, permettant ainsi d'éviter, du moins en théorie, les divergences combinatoires.

Un second modèle, inspiré de la théorie des systèmes généralisés, a été proposé par F. Djindjian (1980). Dans ce modèle un triplet systémique (I, E, S) est défini, ou I représent les informations intrinsèques mesurées sur les objets du système, E les informations extrinsèques exprimant les relations entre les objets du système et les relations avec l'extérieur du système, et S un état de structuration du système d'objets.

Un processus cognitif basé sur l'utilisation de l'analyse des données fait passer le triplet (I, E, S) à un triplet (I', E', S') ou l'information intrinsèque I', l'information extrinsèque E', et la structuration du système S', sont à un niveau explicatif plus élevé :

$$(Io, Eo, So) \rightarrow (Ii, Ei, Si) \rightarrow (In, En, Sn).$$

On remarquera la grande analogie entre les deux approches où Ii correspond à la base de faits, Ei à la base de règles, et Si correspond plus globalement à la proposition Pi.

Si sur le plan du processus cognitif elles sont équivalentes, il faut néanmoins remarquer que l'une est très orientée sur des approches quantitatives utilisant l'analyse des données, tandis que l'autre est orientée vers des approches qualitatives utilisant les systèmes experts.

7. *Difficultés et limites actuelles d'une épistémologie archéologique*

L'insuffisance des données archéologiques

Formellement, la difficulté liée à l'insuffisance de données archéologiques peut s'exprimer de la façon suivante :

$$\exists\ ?\ Po,\ Po \rightarrow Pn,\ \forall Pn$$

Autrement dit, l'échec des constructions archéologiques est-elle simplement due à la faiblesse épistémologique de ces constructions ou, quelque soit la qualité de celles-ci, est-elle directement liée à la qualité des données (Po) ?

De notre point de vue, quelque soit les moteurs utilisés (analyse des données, système expert), qui ont les mêmes forces et les mêmes faiblesses sur ce plan, la potentialité de la construction est directement lié aux données et donc aux processus analogiques.

Le problème avait déjà été posé par Gardin (1979) et Djindjian (1980), à propos des descriptions libres et de leur potentialité. L'insuffisance des données, dans un processus cognitif, conduit à des *structures mélangées* et à des *modèles qui sont toujours validés* comme on a pu souvent le voir en archéologie dans les années 70 et 80.

La multi-interprétation

Formellement, la difficulté de la multi-interprétation peut s'exprimer de la façon suivante :

$$\exists\ ?\ Pi,\ Po \rightarrow Pn$$

Le risque de fausse interprétation d'une structure, ou d'un modèle, ou d'une règle est dû à la trop grande distance cognitive en archéologie entre les données de départ, et les conclusions finales : les structures deviennent cultures, et les systèmes experts divergent.

La faiblesse du processus est liée à l'insuffisance des processus relationnels, et de la localisation des règles qui laissent le système trop ouvert.

La limitation

Formellement, le problème de la limitation peut se poser ainsi :

$$\exists\ ?\ Pn, Po \rightarrow Pn$$

La limitation du processus cognitif peut s'exprimer par la notion de probabilité de validité de la proposition, notion aussi bien valable dans les systèmes experts (mais qui y est généralement faiblement résolue) que dans l'analyse des données (dont les procédures de tests sont souvent discutables).

8. *Propositions pour des éléments de solutions*

L'insuffisance des données

Un travail fondamental, et nouveau, s'avère nécessaire : celui de l'explicitation des analogies et des relations. Son premier avantage sera d'expliciter le rôle des mécanismes analogiques dans la «connaissance» contemporaine en archéologie, et d'en révéler la relativité. Son domaine couvre celui des objets (outils, bijoux, armes, instruments, etc.), des structures (foyers, habitats, villes, urbanisme, etc.), des systèmes (production, commerce, irrigation, administration, etc.).

Le guidage des descriptions par la problématique entraîne la construction de données plus aptes à alimenter efficacement des processus cognitifs : la construction d'estimateurs pour les approches quantitatives, les vocabulaires descriptifs pour les approches non-quantitatives.

L'itération du processus cognitif, enfin, enrichit les données par une simplification syntaxique (réduction des codes) et un enrichissement sémantique (signification plus explicite et plus précise), résultat du mécanisme d'apprentissage, qui améliore la qualité des données en entrée du processus cognitif suivant.

La multi-interprétation

La multi-interprétation est réduite par l'application à chaque étape du processus cognitif de mécanismes de réduction de la variabilité du champ d'application du processus (conditions expérimentales, invariants, répétitions, validations, etc.). Ces mécanismes, sur le plan formel, se traduisent par l'injection dans les processus cognitifs de relations complémentaires, ou de délimitation de validité des règles employées.

La limitation des propositions

Une des faiblesses des constructions archéologiques est de savoir reconnaître et comprendre les limites d'une construction. Ces limites sont dues :

- soit à l'insuffisance des données (par exemple, la détermination de l'âge d'un squelette, en paléodémographie),
- soit à l'insuffisance d'une relation (par exemple, la détermination des acteurs dans l'étude d'un processus de diffusion),
- soit à l'ambition de la construction, dont la faiblesse réside dans l'accumulation successive à toutes les étapes de la construction de données ou de règles insuffisamment probables.

Ainsi les échecs des constructions archéologiques apportent autant d'informations que les réussites. C'est une des raisons pour lesquelles certains spécialistes considèrent que le développement d'un modèle est, en soi, une contribution, méme si le modèle est invalidé, ou plutôt invalidable.

Pour aller plus loin dans cette voie, il y a indiscutablement un changement nécessaire dans la pratique scientifique de l'archéologie qui devrait alors passer d'une ambition des découvertes spectaculaires, à celle d'une ambition cognitive certainement plus empreinte d'humilité.

L'archéologie, par sa nature même, ne peut recevoir une épistémologie prédéfinie d'une science connexe. Elle ne peut qu'élaborer la sienne. L'analyse des formalismes utilisés en archéologie fait émerger des concepts nouveaux ou mal formalisés comme les *processus analogiques*, les

processus relationnels, les *processus d'apprentissage*, et leurs enchaînements. Une analyse de ces processus montre les difficultés et les limites actuelles de leur application, mais fournit quelques voies possibles d'amélioration : l'explicitation des analogies et des relations, la construction d'estimateurs, la relativisation des règles, la publication des échecs et des limites des constructions archéologiques.

Bibliographie

Gardin J. Cl. (1980): *Une archéologie théorique*, Paris, Hachette, 1980.

Gardin J. Cl. (1991): *Le calcul et la raison*, Paris, Èditions de l'EHESS, 1991.

Djindjian F. (1980): *Construction de Systèmes d'aide à la connaissance en archéologie préhistorique. Thèse de doctorat d'archéologie*, Université de Paris 1 (2 vol.).

Djindjian F. (1991): *Méthodes pour l'archéologie*, Paris, Armand Colin, 1991.

ANNE-MARIE GUIMIER-SORBETS*

CRÉATION ET INTERACTION DES BASES DE DONNÉES DOCUMENTAIRES EN ARCHÉOLOGIE

Comme on le sait, l'archéologie a été l'une des premières disciplines humanistes à utiliser l'informatique, en particulier pour l'automatisation des recherches documentaires. En effet, dès les années soixante, des chercheurs réunis autour de J. Cl. Gardin ont posé les principes d'une description archéologique à la fois analytique et combinatoire; les travaux sur la mécanisation de ces analyses ont même été antérieurs à l'informatique, mais c'est elle qui a ouvert des voies nouvelles, et on a vu alors se développer les recherches et les expérimentations sur les deux étapes essentielles de notre travail d'archéologues : la recherche documentaire d'une part, d'autre part l'aide au raisonnement, avec d'abord l'application de méthodes d'analyse mathématique et statistique, puis par des recherches sur les processus cognitifs et plus particulièrement les systèmes experts.

Au point de vue de la technique, on sait combien les outils ont changé: les matériels et les logiciels dont nous disposons maintenant rendent l'utilisation de l'informatique beaucoup plus facile, plus directe et, du moins en principe, plus efficace. Les applications documentaires en ont bénéficié dans de très larges proportions, et aussi des nouvelles possibilités offertes par l'utilisation du mode graphique, l'enregistrement des images, et les nouveaux supports de stockage. Tous ces phénomènes sont bien connus et, pour suivre cette évolution et en tirer le meilleur parti, l'essentiel des recherches a été d'ordre technique : on peut en mesurer les résultats, avec des progrès qui continuent et s'accélèrent.

En me fondant principalement sur l'expérience acquise dans notre Centre de recherche et sur les autres travaux dont je peux avoir une pratique directe, je vais tenter de dégager, comme on me le demande, quelques éléments de bilan et surtout des axes de perspective d'évolution.

* Centre de recherche sur les Traitements automatisés en archéologie classique, CNRS-Université de Paris X.

Jusqu'à présent, les banques de données réalisées sont surtout destinées à des chercheurs et doivent leur permettre d'acquérir et d'explorer des informations. Ce type de produit va évidemment continuer à se développer, mais d'autres types commencent à apparaître qui sont plus directement liés à la diffusion des connaissances auprès de la communauté des chercheurs mais aussi dans la perspective de la formation des étudiants et de l'information d'un public plus vaste.

Les banques de données documentaires sont celles que les chercheurs viennent consulter pour rechercher des informations d'une manière plus ou moins directe selon qu'il s'agit de banques factuelles, dont l'unité est l'objet archéologique, ou de banques référentielles, dont l'unité est la photographie, le dessin, l'article etc.

Pour l'archéologie classique, les banques de données factuelles sont de plusieurs types selon que le champ couvert correspond :
- à un corpus thématique de type traditionnel, et on peut citer en exemple la banque des Archives Beazley réalisée à l'Ashmolean Museum d'Oxford, les banques réunissant des inscriptions grecques ou latines, les banques sur les mosaïques,
- à des collections d'objets conservés dans des musées (la banque des Antiquités grecques et romaines du Louvre et d'autres musées français, etc.)
- à l'enregistrement des données d'une fouille, ou, de façon plus large, des données relatives au patrimoine archéologique national.

Les banques de données référentielles se distinguent selon le (ou les) type(s) de documents contenus dans le fonds documentaire auxquelles elles donnent accès :
- bibliographies générales et surtout signalétiques comme Dyabola (Institut archéologique allemand, Rome) ou Frantiq (Maison de l'Orient méditerranéen, CNRS, Lyon), ou plus spécialisées et analytiques comme la banque sur la religion grecque réalisée par le CREDO à l'Université de Lille III, celles sur l'architecture grecque en cours de constitution dans notre laboratoire en collaboration avec l'Institut de Recherche d'Architecture antique et l'Université de Lyon II.
- photothèques générales pour l'archéologie comme celle du Centre de Documentation photographique et photogrammétrique (CNRS-Université de Paris I) et celle du Centre Camille Jullian (CNRS), ou photothèques

plus spécialisées comme celle que l'Istituto Centrale per il Catalogo e la Documentazione de Rome a constituée sur les peintures et les mosaïques de Pompéi, ou celle du Centre de Recherche sur la Mosaïque (Paris).
- archives rassemblant à la fois des photographies, des plans et dessins, des dossiers de fouille et de correspondance scientifique, comme c'est le cas à l'Ecole française d'Athènes, ou au Comité de Conservation des monuments de l'Acropole.

Toutes les entreprises mentionnées ici ne sont que des exemples de banques de données documentaires relatives à l'archéologie, et on pourrait en citer bien d'autres.

Mais fondamentalement, il s'agit toujours des mêmes produits et des mêmes opérations : ce sont des bases destinées à des chercheurs en quête d'une information de type scientifique; et cette information a préalablement été extraite des objets ou documents par des personnes qui les ont analysés et qui ont exprimé ces informations à l'aide de termes appartenant à un langage documentaire, système de représentation préalablement mis au point.

Cette dernière méthode demeure la plus efficace mais elle est lourde, d'autant plus que l'intérêt des banques est évidement fonction de la quantité et de la pertinence des données qu'elles contiennent. Cette lourdeur fait que si les chercheurs apprécient de pouvoir consulter ces fichiers documentaires automatisés, ils sont moins tentés de participer à leur constitution, du moins si ce travail ne leur est pas immédiatement profitable.

On se trouve alors en face de deux types d'entreprises : quelques banques de données institutionnelles qui, seules, présentent les garanties de durée nécessaires, mais à la constitution desquelles les chercheurs ne participent pas autant qu'ils le devraient; et les très nombreux fichiers informatisés que les chercheurs *construisent* individuellement, pour leur propre étude, qu'ils ne diffusent pas et ne mettent pas à jour une fois le résultat du travail publié, et qui sont ainsi totalement perdus pour la communauté scientifique.

C'est à partir de ces constatations et des tensions éventuelles qu'une telle situation peut engendrer, dans une institution ou plus largement, que nous avons tenté d'apporter une solution à l'Ecole française d'Athènes. Dans cet organisme, en effet, sont constitués divers types de banques,

celle des Archives dont il vient d'être question et à laquelle il faut ajouter le fichier des estampages, les banques factuelles correspondant aux inventaires de dépôt de fouille sur les chantiers de Delphes, Délos, etc., et aussi les banques de type scientifique comme celle sur les amphores de Méditerranée orientale, les monnaies de Thasos, les sculptures hellénistiques de Délos...; et, pour leurs propres recherches, les chercheurs de cet organisme constituent aussi des fichiers qui leur sont personnels.

Ces banques et fichiers diffèrent, on le comprend, à la fois par le type d'unité documentaire (banques factuelles ou référentielles), la nature des données enregistrées et leur finesse d'analyse; ils diffèrent aussi, comme cela est naturel, par leur logiciel d'exploitation et le matériel informatique utilisé (des micro-ordinateurs compatibles PC ou des Macintosh, un mini-ordinateur et des stations de travail sous UNIX). Autre difficulté : ces fichiers nécessitent des polices de caractères grecs plus ou moins riches selon qu'ils ne contiennent que des données en grec moderne ou des données épigraphiques.

Pour permettre une meilleure collaboration en tenant compte de ce contexte très hétérogène, nous sommes parvenus à mettre au point des procédures de transfert d'informations d'un fichier à un autre, de telle manière que les chercheurs nourrissent les banques centrales à partir de leurs propres fichiers et qu'à l'inverse ils puissent les enrichir en puisant certaines informations dans les banques centrales. Ces banques centrales elles-mêmes peuvent échanger leurs données. Ainsi, par exemple pour une inscription, les informations enregistrées à partir de son estampage pourront servir pour la photothèque, la localisation pour les fichiers d'inventaire des sites, et les données d'étude seront enrichies par le chercheur dans son fichier personnel et/ou dans la banque factuelle centrale.

Cette pratique nécessite évidemment un accord sur la nature et la forme de l'enregistrement des données communes ainsi que la mise au point de «filtres» pour le transfert des données d'un logiciel aux autres, mais elle permet de perdre le moins possible des informations collectées et de l'énergie nécessaire à cette tâche qui, sans cela, devrait e re répétée autant de fois qu'on voudrait constituer de fichiers de type différent. Ce qui se fait à l'échelle d'une institution devrait pouvoir se faire plus largement et nous y gagnerions tous.

Toutefois, il faut constater que si la consultation des banques de

données archéologiques devient internationale, leur constitution ne l'est pas encore. Pourtant, là encore, des solutions techniques existent et il est possible, avec certains logiciels documentaires, de réaliser des banques réellement multilingues, c'est-à-dire d'introduire chaque document une seule fois dans une quelconque des langues de la banque, d'interroger l'ensemble des documents dans une langue au choix, et de choisir la langue d'édition des documents fournis en réponse.

Ainsi, la banque de données que nous avons réalisée, avec le logiciel SIGMINI de l'Ecole nationale supérieure des Mines de Paris, sur la Mosaïque dans le monde grec, des origines à la fin de l'époque hellénistique est consultable en français, en anglais et en grec : nous avons déclaré pour cela dans le dictionnaire les équivalents des différents termes dans chacune de ces trois langues; ce travail n'est évidemment pas «fermé» puisque, pour ajouter de nouveaux termes ou même une nouvelle langue, il suffit de compléter les déclarations d'équivalences dans le dictionnaire. Et, à l'inverse, ces équivalences peuvent servir à d'autres banques, qu'elles soient ou non exploitées avec le même logiciel (mais à condition que leur logiciel permette l'établissement de banques multilingues). Les moyens techniques de coopération existent désormais, même dans des environnements linguistiques et informatiques différents, encore faut-il que les chercheurs le veuillent vraiment; et il est préférable que cette coopération ait été prévue dès la conception des différentes entreprises.

Pour la formation des étudiants et l'information d'un public plus vaste, de plus en plus intéressé par la visite de sites archéologiques de musées et d'expositions, on commence à voir apparaître de nouveaux produits multimedia parmi lesquels on peut citer :

- dans la collection des vidéodisques du Musée du Louvre, celui qui est consacré aux Antiquités, et que l'on peut acheter avec ou sans banque de données associée, offre une grande quantité d'images d'excellente qualité.
- le vidéodisque Parthénon, en consultation dans les salles du Musée du Louvre : différent de l'exemple précédent, il s'agit cette fois d'un système interactif plus complet qui offre au visiteur du musée ou à l'étudiant soucieux d'approfondir ses connaissances un choix d'une soixantaine de parcours sur le programme architectural et le décor sculpté du

Parthénon, l'art et la civilisation de la Grèce dans les collections du Louvre.
- l'encyclopédie Perseus, commercialisée depuis peu de temps par la Yale University Press se compose d'un ensemble de vidéodisque et CD-ROM : elle présente un panorama sur la Grèce du Ve siècle destiné à l'initiation des étudiants mais offre également des outils utiles aux spécialistes de cette période. En effet, Perseus permet la consultation interactive de textes anciens dans leur langue d'origine et dans leur traduction anglaise, du dictionnaire Liddell-Scott, d'atlas, et de notices consacrées à la géographie, l'architecture, les formes de vases, la biographie des auteurs anciens, etc. Ces données textuelles sont illustrées par d'importantes quantités d'images, fixes et animées, relatives aux sites et aux objets d'art grec des périodes archaïque et classique.

Le vidéodisque Parthenon comme l'encyclopédie Perseus sont constitués par des chercheurs et destinés non pas (exclusivement) aux autres chercheurs mais à un public plus vaste : cette nouvelle perspective de nos travaux documentaires apparaît tout à fait intéressante puisque, en exploitant les fonctionnalités des produits multimedia, elle permet d'en communiquer les acquis scientifiques à un public élargi.

C'est dans cette perspective aussi que, au sein de notre Centre de Recherche, nous étudions depuis quelque temps les possiblités du traitement documentaire du langage naturel. Certains logiciels, en effet, permettent d'enregistrer des textes et de les interroger de façon libre, par des questions en langage naturel.

Le logiciel SPIRIT de la société Systex, permet la constitution de bases de données textuelles : au fur et à mesure de leur entrée, il assure l'analyse morphologique, syntaxique, et statistique de ces textes qu'on peut ensuite interroger par une question en langage naturel; cette question est elle-même traitée - et analysée - selon les mêmes principes et le système propose les parties de textes qui lui paraissent correspondre à la question posée, ces documents étant classés selon le degré de pertinence calculé par le système, qui explicite en outre les critères sur lesquels sont fondées les différentes classes.

L'utilisateur consulte ainsi tout ou partie des textes proposés, ainsi que les images et les informations factuelles (elles-mêmes interrogeables) qui ont été ajoutées à chaque document. Au cours de cette

consultation, il peut en outre sélectionner un document entier ou seulement une partie, qui devient alors le texte d'une nouvelle question formulée ainsi de façon automatique et qui dirige l'utilisateur vers la consultation de nouveaux documents.

Cette démarche est différente de celle de la base de données «classique» puisqu'elle permet d'explorer des textes écrits pour une consultation traditionnelle, sans qu'il soit nécessaire de passer par la phase d'analyse des informations qu'ils contiennent et leur expression à travers les mots-clés d'un langage documentaire pré-établi. On utilise en fait un autre système de représentation des connaissances – celui du langage naturel de l'auteur du texte, c'est-à-dire, pour nos expérimentations, le texte scientifique non formalisé – et c'est le système très riche d'indexation automatique fourni par le logiciel qui assure la mise en correspondance des concepts contenus dans les textes de la base avec les informations recherchées par l'utilisateur.

Nous expérimentons ce logiciel dans deux perspectives différentes :

- la consultation en langage naturel du catalogue des sculptures hellénistiques de Délos, rédigé par Jean Marcadé dans les années 50 et demeuré à ce jour inédit. La base en cours d'expérimentation rassemblera donc les notices du catalogue, des textes extraits de l'ouvrage *Au Musée de Délos* du même auteur et des images numériques illustrant le même corpus de sculptures.
- la constitution d'un système d'information sur Delphes, l'histoire et la topographie du site, l'architecture, les objets conservés au musée, la religion, les concours pythiques etc. La base rassemblera des textes extraits aussi bien de guides touristiques, des guides archéologiques, des publications de fouille, des monographies et des articles consacrés à Delphes et elle sera illustrée à la fois par une collection d'images analogiques enregistrées sur le vidéodisque «Images de l'Archéologie» et par des images numériques reproduisant des documents graphiques et photographiques complémentaires. Ce système d'information contiendra aussi des documents extraits de plusieurs autres banques de données utilisant le langage documentaire. Il sera destiné à des publics divers comme les étudiants, le «grand public» intéressé, ainsi que les spécialistes.

La mise au point des outils nécessaires à ce type de traitement

documentaire du langage naturel est assez lourde car il faut enrichir les dictionnaires des données linguistiques propres à nos domaines scientifiques mais, si ces techniques donnent un résultat satisfaisant comme cela semble être le cas, on pourra y trouver un moyen d'exploiter des textes déjà rédigés comme les publications, les catalogues raisonnés, les catalogues d'exposition, de musée etc. On peut aussi, et nous le faisons, adjoindre la consultation d'images à celle des textes, et l'appel croisé d'un texte ou d'une partie de texte vers d'autres textes et vers les images qui les illustrent, ou vice versa, permet de simuler un fonctionnement d'hypermedia à liens dynamiques (puisque certains de ces liens sont calculés au fur et à mesure par le système et n'ont pas à être prédéclarés).

Une voie de recherche semble donc être la réalisation de véritables systèmes d'information mettant en oeuvre divers types de données et visant à divers types d'exploitation pour satisfaire divers types de publics : cette hétérogénéité à toutes les étapes du traitement constitue une difficulté supplémentaire mais elle devrait pouvoir enrichir considérablement nos perspectives documentaires. Au delà des problèmes techniques, cette recherche suppose d'abord une analyse aussi précise que possible des objectifs et des publics visés (quelle information pour quel public ?); elle suppose ensuite que nous nous mettions d'accord sur les formats d'échange de nos données (définition de normes); elle suppose enfin - et surtout - que nous soyons d'accord pour les échanger.

Bibliographie

Pour des informations complémentaires, on pourra consulter :

– sur les travaux du Centre de Recherche «Archéologie et Systèmes d'information» : A.M. Guimier-Sorbets, *Les Bases de données en Archéologie. Conception et mise en oeuvre*. Paris, CNRS, 1990. – R. Ginouvès, A.M. Guimier-Sorbets, «Un Centre de Recherche sur les systèmes d'information en Archéologie». *Archeologia e Calcolatori*, 2, 1991, 7-12 .

– sur le vidéodisque «Images de l'Archéologie» que nous avons réalisé en collaboration avec le Centre de Documentation photographique et photogrammétrique (CDPP, CNRS-Université de Paris I), et grâce à un financement du Ministère de l'Èducation Nationale: *Images de l'Archéologie*. Vidéodisque, Paris, 1986.

– sur les systèmes d'information sur la sculpture hellénistique de Délos et le site de Delphes : A.M. Guimier-Sorbets, Ph. Jockey, «Système d'information sur les sculptures de Délos». Communication au Colloque européen Archéologie et Informatique, Saint-Germain-en-Laye, Musée des Antiquités nationales, 21-25 Novembre 1991. Texte à paraître dans les Actes en 1992. – A.M. Guimier-Sorbets, «Using Specialist Knowledge in the Public Domain», Communication au Colloque Data and Image Processing in Classical Archaeology, Ravello, European Centre for the Cultural Heritage, 3-4 Avril 1992. Texte à paraître dans les Actes, revue Archeologia e Calcolatori, Rome, 1992.

– sur la plupart des travaux cités dans cet article et extérieurs à notre Centre : *Traitement de l'information en archéologie*, BRISES, CNRS-INIST, 15, 1989-2 <1990>. A.M. Guimier-Sorbets ed.

MANFRED THALLER*

HISTORICAL INFORMATION SCIENCE: IS THERE SUCH A THING? NEW COMMENTS ON AN OLD IDEA

1. AN INTUITIVE DESCRIPTION

1.1 Introduction

During recent years the computer has been increasingly prominent in many of the disciplines of the Humanities. In the majority of cases, however, this meant just that researchers with a Humanities background discovered that they as well could use tools developed for other people.

This author has for a number of years now proposed, that history would ultimately have to go beyond this stage; that history as a discipline would use data which in the very structure of their informational content would deviate from «information» as it is known in the disciplines dealing with phenomena of current society. We will not repeat the arguments[1] for this line of reasoning, which have been given in the

* Max-Planck-Institut für Geschichte, Göttingen.

1. Most recently: Wolfgang Levermann: *Kontextsensitive Datenverwaltung*, Scripta Mercaturae Verlag, 1991 (= *Halbgraue Reihe zur Historischen Fachinformatik* Band B8). See also among others: Manfred Thaller: «Zur Formalisierbarkeit hermeneutischen Verstehens in der Historie.» In: *Mentalitäten und Lebensverhältnisse. Beispiele aus der Sozialgeschichte der Neuzeit. Rudolf Vierhaus zum 60. Geburtstag.* Göttingen, Vandenhoeck Ruprecht, 1982, pp. 439-454; Manfred Thaller: «Ungefähre Exaktheit. Theoretische Grundlagen und praktische Möglichkeiten einer Formulierung historischer Quellen als Produkte 'unscharfer' Systeme.» In *Neue Ansätze in der Geschichtswissenschaft.* Ed. H. Nagl-Docekal and F. Wimmer. Wien, VWGÖ, 1984 (= *Conceptus Studien* 1), pp. 77-100; Manfred Thaller: «Can We Afford to Use the Computer; Can We Afford Not to Use it?» In: *Informatique et Prosopographie.* Ed. H. Millet. Paris, CNRS 1985, pp. 339-51; Manfred Thaller (ed.): *Datenbanken und Datenverwaltungssysteme als Werkzeuge historischer Forschung.* St. Katharinen, Scripta Mercaturae Verlag, 1986 (= *Historisch-Sozialwissenschaftliche Forschungen* 20); Manfred Thaller: «A Draft Proposal for the Coding of Machine Readable Sources.» *Historical Social Research/Historische Sozialforschung*, 40 (October 1986) pp. 3-46; Manfred Thaller: «The Daily Life of the Middle Ages, Editions of Sources and Data Processing.» *Medium Aevum Quotidianum*, 10 (1987), pp. 6-29; Manfred Thaller: «Secundum Manus. Zur Datenverarbeitung mehrschichtiger Editionen.» In *Geschichte und ihre Quellen. Festschrift für Friedrich Hausmann zum 70. Geburtstag.* Ed. R. Härtel et al. Graz: Akademische Druck- u. Verlagsanstalt, 1987, pp. 629-37;

papers quoted: we would rather more show, that more recent techniques lend additional arguments to it.

To prepare this, we will just as briefly as possible summarize the position presented in earlier contributions. Their our argument has been as follows.

Processing of Historical sources is different from the processing of present day data by a number of reasons: on the most abstract level, this is the case because Historians, when they start their research, do not really «know» with absolute certainty, what their texts mean. Therefore, historical data should be administered in a way, which closely resembles the basic principles of a printed edition as used in the historical disciplines, particularly in the schools of medieval studies. The source itself, we said, could not possibly be wrong: if a name was spelled differently at two occassions this could have been an oversight of the scribe; it could be just as well, however, that this «scribal error» was just the only trace left of the existence of two individuals, separated by a minor difference in the spelling of their names. This being so, we claimed, genuinely «historical» data processing must keep the source as closely to the uncorrected original as possible. Now, six different orthographical representations of one word quite obviously tend to frustrate computer supported analysis; as does the use of currencies of unknown interpretation, complex references to calendar dates and the like. All these problems, however, occur also in printed editions: where in the best ones, from a historian's point of view, therefore two types of information are

---

Computing. Ed. P. Denley and D. Hopkin. Manchester, University Press, 1987, pp. 147-56; Manfred Thaller: «Vom Beleg zum Begriff. Der Beitrag der Datenverarbeitung zur Lösung von Terminologieproblemen.» In: *Ut populus ad historiam trahatur*. Ed. G. M. Dienes *et al.* Graz, Leykam 1988, pp. 237-54; Manfred Thaller: «Gibt es eine fachspezifische Datenverarbeitung in den historischen Wissenschaften? Quellenbanktechniken in der Geschichtswissenschaft.» In: *Geschichtswissenschaft und elektronische Datenverarbeitung*. Ed. K. H. Kaufhold and J. Schneider. Wiesbaden, Steiner, 1988, pp. 45-83; Manfred Thaller: «A Draft Proposal for a Format Exchange Program.» In *Standardisation et échange des bases de données historiques. Actes de la troisième Table Ronde internationale tenue au L.I.S.H. (C.N.R.S.)* Ed. J.-P. Genet. Paris, CNRS, 1988, pp. 329-75; Manfred Thaller: κλειω *3.1.1. Ein Datenbanksystem* St. Katharinen, Scripta Mercaturae Verlag, 1989; Manfred Thaller: «Have Very Large Data Bases Methodological Relevance?» In: *Conceptual and Numerical Analysis of Data* Ed. O. Opitz. Berlin etc., Springer, 1989; Manfred Thaller: «Geographische Angaben in einer Historischen Datenbank.» *Eratosthene-Sphragide* 2 (1990).

carefully kept: the literal transcription of a text and a complex environment of apparatuses and appendices, which contain the interpretations the editor has for the text he presents to the historian using the edition.

This structure, we claimed, would have to be repeated in historical data processing. As a result we presented the basic architecture of the κλειω software system, which carefully distinguishes between the strings of characters administered, which are whensoever possible taken literally from the corpus of source material, and the expert knowledge, necessary to process such data: which are administered by a huge array of dedicated subsystems, applying specific historical knowledge to the data, when they are processed by the computer. So, if two entries in a source, which we have reason to expect to be related to only one individual, refer to this individual once as *Josephus de Mons Friduinus* and another time as *Joe of Montefreidin* both forms will be in the data base; the various representations of historical knowledge operating in the background being responsible for handling the fact that these two forms might actually be just one name. If a document happens to be produced on the *Tuesday after Esto mihi 1513*, that is precisely what enters the data base: that it has been written on February the 8th of that year remains to be computed dynamically, when the software of the DBMS actually accesses such an information.

If we summarize a little bit more abstractly the position we have taken with this approach, we might say in source-oriented data processing, as we have propagated it over the years: *A database contains strings of characters, which are organized for speedy processing. It does not contain assumptions, however, what these strings of characters sysmbolize. To access such a database, it has to exist within an environment of expert knowledge administered by the machine.*

Graphically, we have usually described this situation as in fig. 1.

1.2 New Technologies: Image Processing

In the context of the network of projects, which used these software components during recent years, a lot of work has been done to incorporate image processing as a set of additional capabilities into such a software environment. We try to summarize in the following the four major branches, which we considered as relatively independent approaches to what image processing means within historical research. In all these

cases, we restrict ourselves to the processing of digital images.[2] The following paragraphs assume, that the software environment which is being used contains «fields» of a data type «image», which can be addressed by any query possible within the available query language: so an image can become a candidate for processing because of any combination of values within the other fields it is related to. We assume, that the typical historical research project uses data bases, which handle medium numbers of images: in the pilot projects which are currently being undertaken approximately 20.000 color images (of 2 - 5 MB each) or 20.000 - 100.000 manuscript pages (of 0.5 - 3 MB) are planned to be handled.

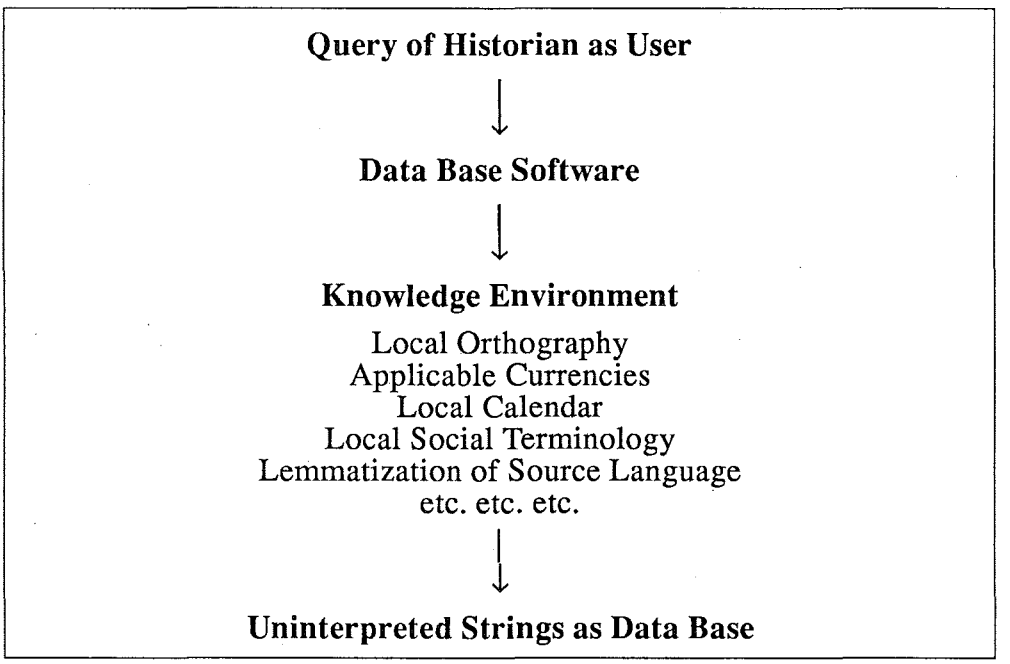


Figure 1: Traditional Architecture of κλειω *Data Bases*

2. Still a good introduction: Rafael C. Gonzalez and Paul Wintz: *Digital Image Processing*, Addison-Wesley, 1987[2].

The experiences from the early stages of these projects can be summarized as follows:

Immediate Image Retrieval. This describes the basic methodology we hinted at, that is, the ability of a software system to display as the result of a query not just the description of an image, but the image itself. To do this within a research environment, a number of practical considerations apply:

- A hierarchical storage administration[3] should be mandatory. This means, that each of the images out of the whole set of 20.000 or 100.000 administered ones can be displayed in the form of a small snapshot without noticeable delay. This is due to the fact, that such small snapshots are being kept on the fastest medium within a storage hierarchy, while the bulk of the image data resides on slower media in the background, access to them possibly even implying the manual mounting of discrete storage media, like magnetooptical cartridges.
- *Each* detail of an image has to be available for zooming with an arbitrary size of the zooming step. This is methodologically extremely important. If you provide images and the possibility to look at only four or five predefined details, the editor of a CD-ROM based system considered to be the most important, the editor is still the only one who controls what you are allowed to be interested in within an image. This does not change dramatically the way of your approach to the images, as compared to a printed book. Only when the user has the possibility to control, what details shall be zoomed – preferably zooming with gain of information – access to the material is better than in the printed solution.
- Of course the usual cutting, pasting, mirroring and similar tools are also necessary within a historical working process.

Image Enhancement. Within the experimental system about 30 statistical

3. In this context it has been implemented for the first time, to the best of our knowledge, within a joint project of IBM Japan and the National Museum of Ethnology at Osaka. See Jung-Kook Hong and Sigeharu Sugita: *A Color Image Database for an Ethnology Museum*, in: Heinrich Best *et al.* (edd.): *Computers in the Humanities and Social Sciences*, K.G. Saur, 1991, pp. 53-60.

operations for transforming and filtering image data have been implemented.[4] These methods – as well as the usual false color techniques – can be applied to any image within the database and or to any segment thereof. The application of these techniques within historical research has usually one or more of the following goals:

- Improving the readability of portions of manuscripts, which became unreadable, because either the writing itself or the material upon which it has been written has changed color, resulting in a reduction of contrast.
- The processing of documents, parts of which have been damaged by mould, humidity or similar reasons.
- The processing of items (inscriptions, coins, etc.) where letters which were cut into the surface or are higher than it were partially destroyed by damage to the object; under some circumstances such material can be partially restored.

Image Binding. Many authors would describe this concept by hinting at «Hypertext» or some of the other «Hyper» concepts. We would like to avoid this, as a matter of intellectual strictness. In data processing there is the concept of nonlinear representations of text: these make up the bulk of this paper, following below. «Hypertext» is a phrase coined by Theodor Nelson, for a very specific and consistent model of a textual data type defined on the basis of specific tools out of the general realm of nonlinear – or nonsequential – texts.[5] This model has so far, never been fully implemented. Application systems like Hypercard[6] are no

4. These techniques are currently realized with the help of an image processing library, *Image Assistant*, which has not been released by IBM for public availability yet, which has been made available to the project, however, by the IBM research laboratory at Winchester by special agreement. At this moment we are testing, whether the general layout of the system is flexible enough to allow for the speedy substitution of this library by a similar – also not yet released – product by Digital Equipment, known as *DECImage*.

5. For reasons which have to do with the very peculiar funding of this project, it is somewhat difficult to give good bibliographic references. Theodor H. Nelson: *Literary Machines* has since 1981 been published in various versions, usually by the author himself. A good summary, which is also easily available: Theodor H. Nelson: *Managing Immense Storage*, in: Byte 13/1, 1988, 225-238; on the concept see also Janet Fiderio: *A Grand Vision*, in: Byte 13/10, 1988, 237-244; see particularly 238.

6. This difference is quite oftenly ignored in the HyperCard literature: a particularly bad example in Carol Kaehler: *HyperCard Power*, Addison-Wesley, 1988, 366-367.

implementations of Hypertext, but systems to administer subsets of the general concept of nonlinear structures.

We therefore prefer to give a more precise definition of our approach, than just a global – and wrong – reference to Hypertext. «Bound Images» we call the administration of bit mapped data objects, which are nonlinearily related themselves, can at the same time also be parts of an arbitrarily complex network of transcribed information, however. This is done in a way, where each portion of the transcription or description in text form, is explicitly related to an area within the bit mapped object. For the sake of completeness we add, that such areas in our implementation may overlap.

The intention behind such a design is the ability to describe in textual form – or by an arbitrarily complex structured description in a factual data base – any object within a manuscript or image. For the practicability of such solutions it is of tantamount importance that the general software system used allows arbitrary and variable degrees of complexity. The examples which have been used for most of our tests are based on descriptions of images, which employ up to eleven hierarchical levels and about two thousand conceptual «fields».

The purpose of these techniques is:

• it shall become possible to search specifically for parts of images. Getting as result of a query *find me a good example of an early modern scabbard* the whole of Altdorfer's «Battle of Alexander» is not what the analytical user of an image is looking for.
• to evaluate the relative position of elements within images. This allows for the systematic processing of questions of the composition of images: e.g., by querying for images, which seem to conflict with the currently assumed canonical rules of medieval painters regarding the placement of motives relative to each other on a picture. In the processing of manuscripts, this allows an analysis of the relative placement of blocks of text, which may help in the differentiation between several layers of writing in a source which has been produced by more than one scribe over time.

Pattern Recognition. Within the projects about which we are reporting here, the application of the techniques which were already mentioned in the section on «Image Enhancement» gave encouraging results in a

number of areas, both when applied to color images and manuscripts. More specifically:

• It is definitely possible, to design sequences of image processing steps, which can be applied systematically to segments of true color images selected by the human using the system. Such sequences are able to reduce the original image towards a representation which shows clearly defined polygons with very few different grey scales.
• The same holds true for manuscripts.
• The polygons resulting in both applications, can be related towards and «measured» against ideal types of basic forms.
• While this is *definitely not* sufficient to clearly identify forms in unprepared images and/or manuscripts, a systematic comparison of the similiarities/dissimiliarities between various sets of images and manuscripts, or their development over time, becomes possible.

1.3 Manuscript Processing – the Next Frontier?

When we try to apply the tools we have just described with regard to images systematically to manuscripts, we can and will aim at the following process in «editing» a manuscript. For the sake of clarity, it should be made explicit, however, that sofar we have described experiences we have gotten with software developed already: the following description is dedicated to our next plans for development.

Step 1: A set of manuscript sources is scanned with medium resolution. According to earlier experiences with the integration of WORM disks and magnetooptical devices into traditional workstations, we can assume that the administration of 10.000 - 20.000 pages of manuscript on PC's of the upper or workstations of the lower range will be fairly easily possible within two or three years from now. For a precise definition of the capacities which we can achieve, we do not have sufficient experiences with the speed possible in scanning such sources with a satisfactory quality.

Step 2: The documents which in that way are preserved permanently in photographic quality, are loaded into a data base: this data base contains at the beginning just a series of document identifications (basically archival numbers) and the scanned images of these documents.

Step 3: When working with such sources, the historian first of all transcribes the manuscripts literally. For this purpose the following tools are at his command:

- By a system of graphically displayed reference coordinates, the historian can bind individual transcribed phrases (or important abbreviations, or individual letters significant for the specific scribe) to the transcription of these portions of the text.
- Is a reading difficult, it is possible
  - to improve it by the means of the image enhancement techniques discussed.
  - but also to get a set of comparable portions of the manuscript transcribed so far. Either by asking for the graphical representation of cases where the possible readings have been transcribed at earlier stages of the working process or asking for cases where such transcriptions are within a given degree of similarity to the graphical form to be transcribed now.

Step 4: As soon as a continuous transcription exists, or parallel to its creation, tools exist to insert into the text symbolic markup, specifying e.g. persons mentioned or topographical entities referenced. While in the design plans of our project a more general notion of markup is employed, *specifically emphasizing that there are situations, where the graphical representation of symbols may be important,* SGML (*Standard Generalized Markup Language*)[7] would be a good example for a fairly general markup language which could be employed for such purposes. While we would like to stress, that we consider the question of how such markup schemes should be constructed for other than printing purposes to be anything but closed – rather more: not even opened up – one should point to the efforts of the *Text Encoding Initiative*[8] to define common

7. Charles Goldfarb: *The SGML Handbook*, Oxford University Press, 1991; vgl. Lou Burnard: *What is SGML and how does it help?*, in: Daniel Greenstein (Ed.): *Modelling Historical Data*, Scripta Mercaturae Verlag, 1991 (= *Halbgraue Reihe zur historischen Fachinformatik*), pp. 65-79.

8. C. Michael Sperberg-McQueen and Lou Burnard (Edd.): *Guidelines for the Encoding and Interchange of Machine-Readable Text*, Chicago and Oxford, Draft Version 1.0, 1990.

rules for the applications of such markup schemes in specific areas of the Humanities.

In our context such markup is instrumental in converting the transcribed text into a component of a structured data base, into which each portion of the transcription is supposed to be parsed immediately after markup has been completed.

1.4 How do those things interrelate?

We have started with a description of why we have during recent years argumented for a specific architecture of data base systems in historical research; we have emphasized, that they have to contain a considerable number of tools to apply specific knowledge to transcriptions of text, the precise meaning of which changes during a project. The more recent developments we described very sketchily above fit into such an architecture very well: actually we expand the model just in two ways. On the one hand, we assume, that such a system is now able to refer from a character-coded transcription to a bit mapped representation and back; on the other we assume, that the knowledge administered by the system as a whole now contains additional items of knowledge, e.g., the set of «ideal forms» of the letters peculiar to a specific scribe. Schematically, we therefore get the result displayed as figure 2.

There remains the original aim of this paper. We claimed, that the line of reasoning, which brought this author for a number of years to the assumption already, that the use of computer techniques in the Humanities would need more than just the application of commercial packages, would get further support, if we consider more recent technologies. Here the type of our argument, unfortunately, has to change abruptly: while so far, we more or less tried to describe fairly intuitively, how such technologies may influence our current work, we can prove the need for more systematic developments only, when we try to enter a serious technical argument. The remainder of this paper is therefore dedicated to an attempt to define a concise implementation of a non-linear data type «text» for a Humanities' – and specifically – Historian's application system, which fully supports the closely bound mixture of transcribed text and bound images. What we describe on the following pages, is a fully nonsequential data type; it is not hypertext, however, but an alternative proposal for a data type «text».

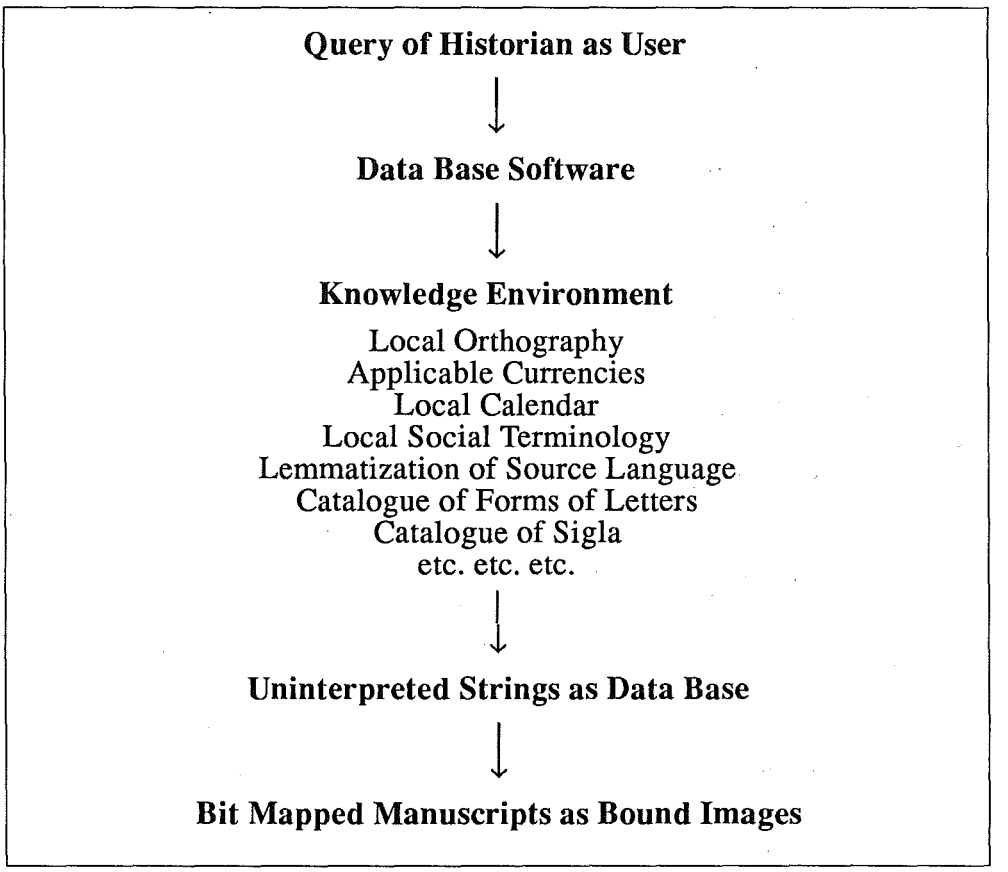


Figure 2: Enhanced Architecture of κλειω *Data Bases*

2. DESIGN PROPOSAL FOR A NONLINEAR DATA TYPE

2.1 What makes a text «historical»?

Speaking on the most general level, we consider a text to be «historical», when it describes a situation, where we do neither know for sure, what the situation has been «in reality», nor according to which rules it has been converted into a written report about reality. On an intuitive level this is exemplified by cases, where two people with the same graphic representation of their names are mentioned in a set of documents, which possibly could be two cases of the same «real» individual being caught acting, which, however could also be homographic symbols

for two completely different biological entities. At a more sublime level, a change in the color of the ink a given person uses in an official correspondence of the 19th century could be an indication of the original supply of ink having dried up; or of a considerable rise of the author within the bureaucratic ranks. Let us just emphasize for non historians, that the second example is all but artificial: indeed the different colors of comments to drafts for diplomatic documents are in the 19th century quite often the only identifying mark, which diplomatic agent added which opinion.

What these introductory examples should demonstrate, is, that the text – the computer interpretable representation of a written document – forms in historical research an intermediate layer between two other layers of information. On the one extreme we have abstract factual knowledge about the various entities described in a text, which allows the interpretation of it; on the other there are purely graphical characteristics of the written document, which may carry meaning, but need not do so.

That the second problem is a genuine markup problem is probably obvious: if we use a computer to prepare diplomatic drafts of the 19th century for printing, we obviously need a way to describe a portion of the document as being «written with blue pencil». Which, at the time of the first transcription is exactly what it says, a literal description of a graphic property, though during the process of research it *may* well acquire a more abstract connotation, like `author=M. Simpson`. This could of course be interpreted as such properties being eminently fitted to abstract rules for markup, because at the time of producing the markup we have not yet the faintest idea what the final representation in print, if any, of the specific graphic property is to be. Quite besides that at least I cannot very well see, how it should become possible to propose a finite list of such potentially significant graphical properties, there is a more basic problem. We all the time have now been speaking about graphical properties which *may* represent some meaning. Which is another way of saying, that this graphic property is purely accidental. To bring it to a point: almost all the examples given in the discussions on standardization during the last few years dealt with how to tag a structure which is clearly understood and where the graphic representation is accidental. Historical work deals with structures in a text which we want to *discover*, where the graphics we see may be all the clues we ever might get.

The real difficulty behind this might be a somewhat imprecise definition of what a «text» is to begin with. To me it seems, that in almost all the contributions to relevant discussions, a «text» is seen as either the starting point for research or the result of research: either what you get from a colleague to make your linguistic, stylometric or whatsoever analysis from or what you are going to deliver to the printer and, potentially, at the same time to another colleague. In the understanding of this document a text needs to be a considerably more dynamic kind of thing, the formally treatable representation of the current assumptions of a researcher about what his documents actually contain.

This on the one hand means, that we have to provide facilities to mark up graphical attributes which *may* acquire substantial meaning; on the other there have to be provisions for a link between a text and a set of assumptions about portions of it. To go back to our initial example: when we provide for marking a portion of a text as representing the «name of a person», we will also have to provide for a link to some background data base, which contains descriptions of «real» persons, being represented in the text by all kinds of conflicting graphic representations. State of the art data bases in history actually carry this a step further, by providing separate links between the graphical variation of a name to an algorithm, which is supposedly able to filter out the «accidental» orthographic variation of the name, before it is being linked to factual knowledge about the person this name is a tag for.

So, a historical text is in this document considered to be a representation of assumptions about some historical reality, containing on the one hand descriptions of graphical properties, which may require interpretation, at the other linkages to representations of knowledge, which are connected to a specified portion of the text.

This simple model has, however, to be extended into two directions. A «historical» text is in our opinion something which has come to us under a consistent set of circumstances: our interpretation of colored annotations can obviously be valid only within a corpus of materials which came into existence within one bureaucratic unit. Similarly the language of a medieval chronicle can be analyzed, at least in the first step, only within one copy of that chronicle, though it may have been transmitted, with minor variations, in a whole family of texts. At the same time, however, the reality described by the process of formulating a

political document can very often be understood only, if two parallel sets of comments upon some drafts, by different branches of the bureaucracy, are interpreted synchroneously; and the «story» told by a medieval chronicle can only be analyzed, if it is seen as complete as possible: though sometimes no single text exists, which contains all the parts of it.

By first approximation this means, we need a mechanism, to administer an integrated document as an entity, which consists of several layers of traditions, each consisting of some «text» – i.e. a collection of words – which can only be interpreted in the context of some assumptions about the rules applicable to it. As, obviously, for some portions of the fictitious «true chronicle of x» conflictingly tradited texts will exist, this leads directly to the requirement of a text representation which allows a given portion of text to have more than one equally valid form. We have, therefore, also to provide for a mechanism, which allows a dynamic handling of variants, which enables software, to treat one coherent representation of a text on a computer, as if it would just consist of one manuscript as well as if it would consist of the logical sum of two or more manuscripts. The computer representation of a machine readable text should therefore in our opinion not only make it possible to handle variants, but to treat all streams of tradition combined into a «text» as potentially equal.

Finally, we have to define the relationship between a «text» as a running representation of a tradited document and a «text» as converted into a database according to some abstract model. In our opinion these two representations should be seen as very close to each other, allowing the database to inspect the natural language context out of which its entities of attributes have been derived, and on the other hand allowing the user of the machine readable text to jump from one portion of it to another portion which, irrespective of the language used, deals with the same abstract concept. More pragmatically: if you enter into a historical database a query like «When did the monastery of St. X receive more than five solidi from a single tenant?» we want to see at least the unstructured description of the relevant entries in the administrative records, if not the scanned image of the respective page, and when we encounter in our running text a peculiarly verbose eulogy about a given benefactor, we would like to be able to get all other sources related to that benefactor, be they in the same source or not.

We are quite aware, that the requirements we just described can be met only in part by existing software. We think however, that there is small, if any, sense to concentrate completely upon the task of how to code data in such a way, that the can be handled by present day software. As a matter of fact, tagging a text exhaustively and completely seems to create such an additional overhead, that I see spurious chances at best, that any historian could be convinced to enter all needed tags by hand. So what we define in the text representation committee is certainly no markup, which normally will and shall be entered by a historian: to pretend otherwise would in my opinion be nothing but fictitious. The sense of a standardized text representation at least in historical research – and decidedly there – can only be to create a means for the communication between software systems, not between human historians.

As such, however, we need a medium that does take account of things to come and is broad enough to give software designers some reason, why they should invest into implementing components, which support such recommendations as an exchange format. To do so we need some foresight; which is why we start from a non-existing system.

2.2 A «Historical Text Engine»

To allow us to do all the things specified in a coherent computing environment, we would first of all like to sketch how such an environment should behave.

We asume on the following pages, that all texts are treated as «information strings». A running text consists simply of a collection of linearily ordered strings of this type; a data base or knowledge representation consists of texts which are connected in a non-linear way. As every linear structure can be described as a trivial case of a non-linear one, running texts, (factual) data bases, full text bases, knowledge bases and, as we will see, collections of bit-mapped data objects are all to be considered as specific realizations of a general representation of information. To make that possible, we assume further, that all the necessary string handling operations are taken care of by a «text engine» which relies on other software components to be provided with correct «information strings» irrespective how they are administered. We will see, however, that links to other «information strings» can be part of any of them.

In any implementation of the following concepts, a «text engine» could therefore be only realized in close connection with other dedicated software systems, which take care of administering the relationships between various information strings. These do not form part of the present considerations. As the definition of the various items of information to be handled requires references to them sometimes, we will, however, just shortly define the three most important tools of that type.

In our concept we did stress the similarity between a running text and a structured data base; indeed we will later see, that we are also considering cases, where one collection of information strings can alternatively but synchroneously be interpreted as a running text and as a data base. To make that possible, we assume that besides the text engine, which we cover here the following exist.

A *text administrator.* This is a very primitive program, which does not very much more, than performing I/O on strictly sequentially stored collections of information strings. A text processing system in our concept would use such a text administrator to save and load information strings from background media, which then are processed with the help of tools from the text engine. Whenever we use the term «textprocessing» in the remainder of this paper, we refer to software, which performs typical tasks of current day textprocessing, including primitive full text retrieval applications, by using the services of both, text administrator and text engine.

A *data base engine.* This is a family of software tools, which are responsible to administer non linear collections of information strings. These software components are responsible for the handling of all problems resulting from the adaptation of current retrieval concepts to handle context sensitivity of queries and uncertainty or ambiguity of structural relationships.

A *knowledge engine.* This is a family of software tools, which are responsible for the administration of all such conversion and transformation processes, which are built upon knowledge as are based upon dictionary-like structures or complex sets of rules. As all information is supposed to be evaluated dynamically, these software components in turn use components of the text engine, when the need to handle information strings arises.

These «information strings» which are treated by our assumed text

engine in the environment shared with these other major modules, consist of linked lists of «uninterpreted items», which exist in an «interpretative environment». Whensoever an information string is handed to the text engine, it is guarantueed, that the later is supplied with a full copy of an interpretative environment.

While more precise definitions of interpretative environment and uninterpreted items will be given shortly, it makes their respective roles probably easier to understand, when we describe them somewhat intuitively first. As a first approximation, we could consider the interpretative environment as a table of mappings of abstract font commands into concrete printer operations. So when we look at the text engine operation «prepare output on a specific printer» the printing of the string starts with all such applicable parameters regarding printing and spacing, as can be derived from the interpretative environment handed over with the string to be printed. After this, the uninterpreted items are inspected and item by item converted into such strings and/or printer commands as represent their output form in the light of the current interpretative environment. While this is a description of a current day printing process, in our opinion it should get further: the «font» of a text not only being relevant, when it is being printed, but also, when a string in font «A» is compared to a string in font «B».

A very important consequence of this separation between uninterpreted items and interpretative environment has however to be clarified already now. As mentioned initially, we deal not immediately with the question of markup. We assume, however, that the internal representation of a historical texts indeed needs some features, which are inherently like a symbolic markup: i.e., some information about how the text shall be processed, which is interpreted only, when the text is being processed. This produces a subtle difficulty, when we are speaking about non-linear structures of text, where individual parts of the text shall be accessible. As we will see further, the interpretation of character i of a text may depend on some information, that is contained between character i - 100 and i - 90 of that text. So, if we want to interpret the i^{th} character correctly, we would have to know, that information relevant to that character occurs before it in the string. Therefore we assume, that a «string of information», as we define it, is always administered so, that it is only accessed at a point, where it can be guaranteed, that all information

necessary for its interpretation is available. More formally, we speak of *entrance points* into a collection of strings, where a complete copy of the interpretative environment for the following character is available. All characters between two entrance points can only be correctly interpreted, when the text engine reads and interprets first all parts of the string of information, which are situated between the nearest entrance point and the character in question.

The importance of this concept can scarcely be overestimated. Indeed, the need to provide a sufficiently but not unnecessarily large number of entrance points, is the main reason, why we distinguish so sharply between a strictly sequential and linear text administrator, a strictly non linear data base engine, which, however, can assume, that from each of its items a path to the nearest applicable entrance point is defined, and a knowledge engine, which handles dictionaries of relatively small information strings, each of which has its own entrance point, as the can be accessed completely at random.

2.2.1 Types of uninterpreted items

Strings of uninterpreted items are made up of five different classes of items:

- *Basic items.*
- *String qualities.*
- *String links.*
- *String variants.*
- *Embedded structures.*

The role of these classes of constituents are in turn:

2.2.1.1 Basic items

These items carry the actual information derived from a historical source. In the most trivial case, the consist of simple character codes. *All* such items, however, are considered to have logically the same rank. That is, a small bit map (e.g. for a non-deciphered language like the Indus hieroghlyphs or a non-textual symbol, like a water mark in paper) or a plain ASCII character can both form distinct items of a «string» in

our sense. This assumes, that the text engine contains tools, which can sort and compare *all* types of basic items.

The following types of basic items are defined:

- *Simple characters.*
- *Character tokens.*
- *Bit mapped tokens.*
- *Pictures.*

2.2.1.1.1 Simple characters

Simple characters are described by a sequence of *n* bytes per character, *n* defaulting to one in most text engines. It is assumed in this paper, that characters which represent letters, have one case only. For reasons which are given further below, it is assumed that case is just another string quality which does not justify a special treatment. Each simple character is represented by a numeric value, which indexes a table that contains a variable amount of information about the character. That information consists of:

- Sorting position of the character within the table.
- 'Binding' of the character. By this property we define its behavior in conjunction with neighboring items to it's left and right within the same string.

2.2.1.1.2 Character tokens

A character token is represented by a traditional – henceforth called primitive – string of simple characters; in most real-world application starting with a common escape character. While being represented by a primitive string, the are conceptually, however, just the same as simple characters: the degree of similarity between two text tokens – as, e.g, expressed by a table of sorting values – is therefore completely independent of the string representation of the tokens. As the two primitive characters «a» and «A» may or may not be considered identical, independent of the code values assigned to them, in a historical text the two text tokens «`\chrismon`» and «`cross`» may or may not be considered to be identical or similar; there is, however, no inherent relationship created by both tokens starting with the primitive string «`\c`».

2.2.1.1.3 Bit mapped tokens

Bit mapped tokens are tokens, for which all is valid, what has been said about the properties of text tokens. Bit mapped tokens do not consist of a sequence of primitive characters, however, but of a sequence of the form: `escape-character-length-bitmap`. A further difference is, that their similarity is defined not by a tabular listing of their relationships, but the decision rules for the compariosn of the bitmaps themselves.

2.2.1.1.4 Pictures

Intuitively pictures are obviously the same as bitmaps: indeed, their internal representation is assumed to follow the same rules, as just given in the preceding section. While a bit mapped token is assumed to be an atomic item of information, a picture is assumed to be a possibly structured entity, which may occur as part of a text, will more often be connected to it, however, by the mechanism describe in section 3.1.3.5 for text links.

2.2.1.2 String qualities

As mentioned initially, each uninterpreted information string exists in an interpretative environment. This is defined by a number of assumptions, which are true for the first information of the string. The information string contains, besides the basic items discussed sofar, which carry the «real» information, indications for a change in any of these assumptions. This implies for the text engine, that all of its constituents are guarantueed to start the processing of a string only at well defined starting points, all operations defined on the strings parsing allong them. While this may seem to be a distraction, we would like to emphasize it here, as otherwise the concept of string quality cannot be understood.

Every string of information exists in an environment which defines its

- *modes*,
- *style*,
- *color*,
- *size* and
- *view*.

It should be noted here, that these names have been choosen for intuitive plausibility, as have the examples below. The flexibility of the concepts, however, is to be derived from the abstract definitions given.

2.2.1.2.1 String modes

String modes define the absence or the presence of a set of attributes. That is, a given item of information can have an attribute or can miss it. It is not possible, to have a mode in a certain degree. Every string of information inherits from its interpretative environment a set of default modes. If a certain mode is not defined in the environment, it is assumed to be absent.

The most intuitive example of a string mode is the case of a character. As we defined before, that simple characters are assumed to be caseless, it would completely depend on the interpretative environment, whether the string

```
this is a string
```

would be interpreted by the text engine as upper or lower case. By interpretation we mean in this and all following examples, the behaviour of *all* components: an «upper case» string would be printed as upper case (if possible on the output device) but its being uppercase would also influence comparison operations (see below).

At any point in a string a mode can be activated or deactivated:

Mode: +case this is a string

would always result in an uppercase string, irrespective of the assumptions of the interpretative environment,

Mode: -case this is a string

always in a lowercase one. The interpretation of

Mode: +case t *Mode*: -case his is a string

is clear.

As each mode, which is not explicitly defined in the interpretative environment, is assumed to be absent, their number is arbitrary and has not to be known by the text engine. Modes which are encountered in an information string, for which the text engine has no explicit instructions are therefore completely ignored. In the case of

Mode: +case t *Mode*: -case his is a *Mode*: +german

Mode: +case z *Mode*: -case eichenkette *Mode*: -german

the mode «german» would in most search operations be ignored; in full text or data base applications it could, however, be used as a selection criterion irrespective of the structure which defines the relationship between this information string and all others in the currently administered data; in Anglosaxon text processing applications it could be interpreted as underlining.

2.2.1.2.2 String style

While any item in a string of information can at the same time have an arbitrarily large number of modes, it alway has precisely one style. Statistically speaking, the style of an item is handled on a strictly nominal level: there are no assumptions about any relationship between two different styles expressed in the internal representation of styles.

The most intuitive example for the style of a text would be font information, as in the example

Style: german zeichenkette *Style*: basic

Please note, that this is not exactly the same than the previous example: while in the previous one, «z» could acquire the mode `case`, without loosing the mode `german`, no part of `Zeichenkette` could acquire the style `gothic` without loosing the style `german`.

2.2.1.2.3 String color

The quality of color is similar to that of style, by its values being mutually exclusive. Its intuitive interpretation is probably obvious and the introductory remarks of this paper show a potential application. For a systematic interpretation, however, it is much more important, that this

quality is supposed to represent statistically an ordinal level. That is, it is assumed to be represented internally by ordinal numbers, which allow expressions of similarity. (A similarity to the implementation of the `enum` concept in the 'C' programming language is intentional.) The two strings:

Color: dark blue George Smith

and

Color: light blue George Smith

would in most interpretative environments therefore be assumed to be closer to each other, than the strings:

Color: dark blue George Smith

and

Color: light red George Smith

It would, however, *not* be possible, to express the difference in the degree of similarity between the two pairs of names. (This is a statistical statement and a definition of the concept of `color`, not a statement about artistic and/or biological perception of colors.)

2.2.1.2.4 String size

String size, too, has a pretty obvious application. It is similar to the concept of color, allows additionally, however, to express a difference in the degree of similarity between two strings of information being compared. In the three fragments:

Charter a: *Size*: 20pt \chrismon *Size*: 10pt In nomine individuę trinitatis ...
Charter a: *Size*: 30pt \chrismon *Size*: 10pt In nomine individuę trinitatis ...
Charter a: *Size*: 40pt \chrismon *Size*: 20pt In nomine individuę trinitatis ...

the chrismon in a is more similar to the one in charter b therefore, than to the one in charter c; the proportion between the sizes of chrismon and main body of script, however, is identical between charters a and c, while both are dissimilar to b in precisely the same degree.

2.2.1.2.5 String views

The qualities so far – with the possible exception of color – may be seen as an attempt to define classical typesetting attributes in a sufficiently systematic way to allow their interpretation on an intermediate level between typographical representation and conceptual understanding. We recapitulate: the color of a note in a diplomatic document *may* ultimately acquire some meaningful, abstract interpretation; at the beginning of an editorial process, however, it will be exactly what it looks like: proof that Mr. X used a blue pencil.

The concept of *string view*, on the other hand, has been introduced to handle phenomena, which often occur in manuscripts, have, however, no generally accepted typographical conventions assigned to them.

Typical examples would be portions of a text, which are legible and obviously part of the original manuscript, but which later have been crossed out, additions being added at the same time, or manuscripts, which have been written by a number of scribes, some of which can be identified, while others can not clearly be distinguished. Obviously all these properties of a manuscript could in principle be covered by the string qualities given so far.

The tools provided so far, did always assume, however, that each of the qualities would exist alone: a set of binary qualities, exactly one nominal quality, exactly one ordinal and exactly one which allows comparisons of degrees of similarity. To generalize this model, we introduce the concept of a *text view*, which is defined as *View:* `Type, Name, n`. As its first argument, it accepts any of the previously identified text qualities, i.e., `mode, style, color` or `size`. It introduces a named text quality, which has the properties discussed so far. So our previous notations could be seen as shorthand for a more general text view notation. The following equivalences would hold:

Mode: n == *View*: mode, default, n

Style: n == *View*: style, default, n

Color: n == *View*: color, default, n

Size: n == *view* size, default, n

The difference between these two definitions is, however, more important, when it comes to actually implementing such a model. We assume, that a text engine optimizes the four *default views* with regard to speed of processing of individual information string. This means, that when one of the default views is encountered in an information string, it will be taken care of by an extremely quick operation. When an explicitly named view is encountered, however, the text engine is allowed to reorganize the interpretative environment to allow for it. (All comparison operations have to allow for `size` sensitivity without loss of efficiency; a comparison which has to allow for five independent sensitivities for `views` of type `size`, however, is allowed to be significantly less efficient than a comparison that handles just the default `size view`).

This differentiation – and more so the space it is assigned – may be a reflection about the author's background in actual program development: we consider this differentiation to be extremely important, however, as, on the other hand we assume, that historical texts can be handled correctly only, if the number of `views` allowed is unlimited.

2.2.1.3 String links

While we consider string qualities to be a more systematic description of classical textual properties, string links define the conditions for assembling individual information strings into larger objects, like texts or data bases. Basically we consider it necessary to embed into a string reference points, from which it is possible to branch to other strings. The intuitive example for this would be a footnote.

As we mentioned initially, we consider a text not so much to be something which primarily has to be printed, but as a representation of the current knowledge about some historical phenomenon. All such points, where it shall be possible to branch from a given point of reference within a text to somewhere else, are therefore meaningful only as being connected to specific operations of the assumed text engine.

These operations are:

- *branches*,
- *text references*,

- *data base references,*
- *knowledge references* and
- *bitmap references.*

2.2.1.3.1 Branches

A branch is the most simple string link. It consists of a pair of addresses, connecting an arbitrary point within a string of information with the entrance point into another string. The intuitive example for it is a note in textprocessing. Branches pointing from an arbitrary point of an information string to the entrance point into another string, we will call *exceptions* and denote with the symbol `Name→` ; branches from the entrance point of a string to an arbitrary point of another information string, we will call reference and symbolize by `←Name`.

In the case of a footnote, these elements would be used as follows:

```
footnotes →  this point is usually not discussed
any more ...
← footnotes  Cf. John Smith; ...
```

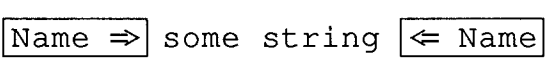
The text engine resolves the arrows in these string links as follows:

- exceptions are plain pointers to the beginning of another string of information, allowing the interpretative environment to be initialized the standard way.
- references are similar pointers to the arbitrary point from which the exception did branch away. They can, however, only be traversed, if this point in the collection of strings has been reached via a previous refernence from the corresponding exception. In such cases the text engine stacks a copy of the state the interpretative environment has been in, when the exception was activated. If the reference is reached by any other navigational operation within the collection of strings in question, it is not possible to follow it to the spot of the exception.

2.2.1.3.2 Text References

Text references allow it to bracket a specific portion of text and logically to assemble all such portions into a specific collection of texts. An intuitive example within text processing would be the creation of registers.

Text references consist of pairs of the form



which we shall discuss as «forward reference», «reference string» and «backward reference» respectively.

A *forward reference* consists of

- a mark pointing to the end of the reference string,
- a pointer to the next forward reference in the collection of strings with the same `name` and
- a pointer to the nearest entrance point into the information string containing the next forward reference in front of it.

It enables a text engine therefore, to navigate from one text reference immediately to the next; does not remove the necessity, however, to interpret the portion of the information string in front of the respective forward references to bring the interpretative environment into the state it has to be, when the reference string shall be interpreted correctly.

A *backward reference* contains the same information, does so with respect to the preciding text reference in the collection of information strings in question, however.

2.2.1.3.3 Data Base References

Data base references have no precise equivalent in traditional applications. A specific data base pointer  is defined by

- the data base which shall be referenced,
- a procedure specifying for the data base engine in question, how the reference shall be converted into an information string.

When control returns to the text engine, a modifiable copy of the information string created in this way is brought to its disposal and for purposes of textprocessing integrated transparently into the «text» proper. Obvious, but trivial examples of usage would be the dynamic treatment of bibliographical and/or biographical information.

2.2.1.3.4 Knowledge References

Knowledge references are denoted by identical bracketing reference symbols of the form

↓KB object text ↓KB

Object text refers to an information – like complex chronologies, historical currencies and similar, as described in previous papers by this author – which needs to be transformed, before submitted to specific classes of treatments. (Think once more of sorting or comparisons.)

The *knowledge* referenced is assumed to consist of a formal definition of the format a object text has to have to be processed plus a collection of suitable dictionaries to apply specific transformation rules.

A specific ↓x consists of

- a specification of the knowledge (base) to be accessed plus
- a set of operations of the text engine, which trigger the transformation of the object text.

For all such operations, the text engine does not process the object text itself, but the result of the conversion.

2.2.1.3.5 Bitmap References

A bitmap reference consists of a pair of identical bracketing symbols of the form

‖x interpretable equivalent ‖x

It consists of

- a reference to a bit image, which either is administered as a «picture» as defined above or as a completely independent file and
- a set of coordinates within the bitmap.

The text engine references the bit image, when suitable display units are available, uses the interpretable equivalent, however, when functions are being called, which require operations like search or comparison.

2.2.1.4 String variants

String variants have an obvious application: the administration and processing of the apparatus criticus of a text. The following model for the integration of variance into a general text representation model sees this only as a starting point, however. We rather assume as application the creation of dynamic text representations, i.e. of text representations, which allow software to treat all variances of a text as «equal». This could be envisaged by display modules, which allow the user to switch from variant α to variant β by hitting a function key, the displayed text in all cases where variants exist in both stages following the readings of manuscript α or β.

It is assumed that – unless indicated otherwise by the interpretative environment – an information starts with a part that is present in all witnesses for a given text. This assumption is changed, as soon as in the processing of the information string a *block border* is encountered. Block borders, which can be nested, limit parts of an information string, for which substitutable variant readings exist.

Generally a text with variants would therefore be represented as

`text present in all witnesses` `Block≫` `variant readings` `≪Block` `text present in all witnesses`

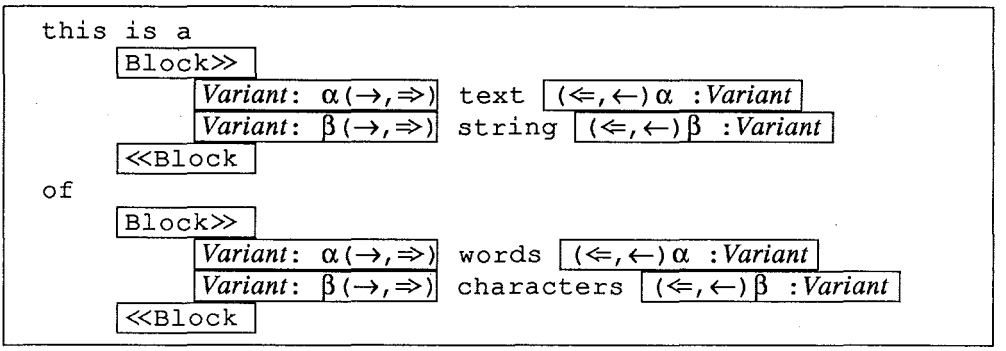
The *variant readings* consist of blocks of the form

Variant: `Name (→,⇒)` `text of reading` `(⇐,←) Name` *:Variant*

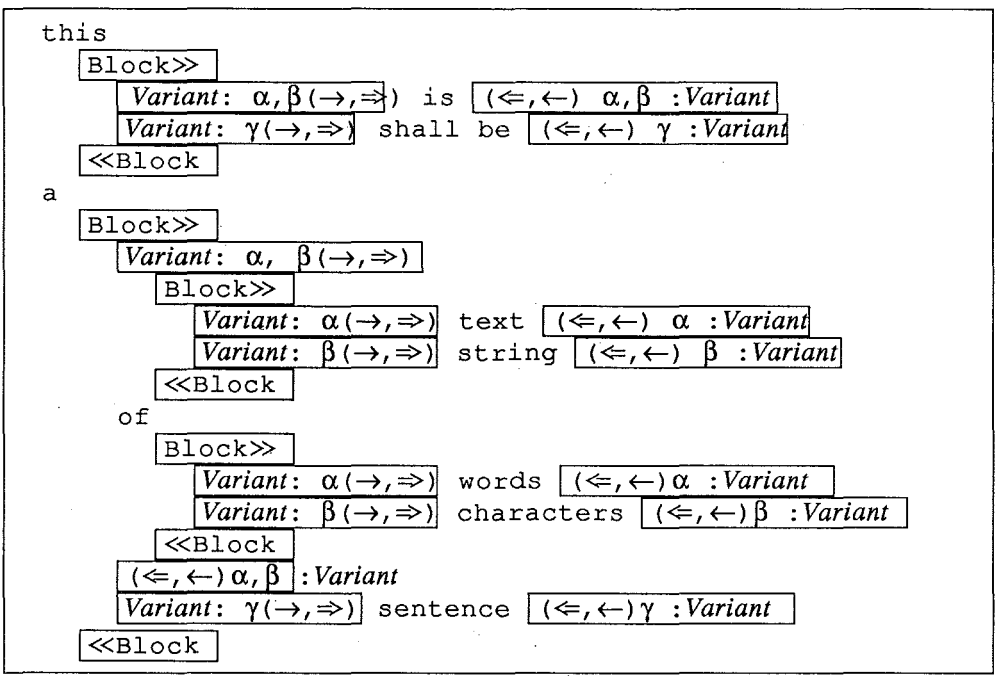
where *nameset* specifies in which witnesses a given reading is present.

The (→, ⇒) and (⇐, ←) symbols indicate, that all variant readings are linked as a list, where each atom has a pointer to the next variant reading, but at the same time also a pointer to the end of the respective block to speed up processing.

The mechanism is probably best explained by two examples. Lets first look at the situation where manuscript α gives `this is a text of words`, while manuscript β reads `this is a string of characters`. In this case we assume a representation which is:



When, to show nesting, we add manuscript γ which reads `this shall be a sentence`, we get:



As this may look somewhat complicated: let me remind the gentle reader, that we are describing here a structure, which internally shall be able to handle something. Entering markup into a text, which in turn is parsed to provide the required pointers, would be one way to achieve this.

2.2.1.5 Embedded structures

While the preceding parts of this paper form presumably a general model for the representation of historical texts, this section may be more specific to the activities of the author. It deals with the problems from representation, where a data base, which follows a network model, is integrated into a text. The idea is, that data base queries are processed, which are translated via structural information contained in a data dictionary into such navigational processes as are required to select various items of information for processing. Unlike in traditional data bases, that information is, however, not «extracted» from a data base. Instead of extracted information from natural text, we assume, that the text as a whole is represented on the machine, various pointers – similar to the classes of such, which we described sofar – being «hidden» within the running text, which define that a given part of it is not just a series of characters, but at the same time the content of a structurally defined field of a data base.

The idea is, that we have a text, like the sentence *On the 19th of March 1764, John Winslow and George Bilmington appeared before this court to claim* .... which by software based upon our ideal text engine can be handled for typical purposes of textprocessing, other software based upon this engine can at the same time, however, treat terms contained within it – like *John Winslow* – as the value of the attribute `name` of an incidence of the entity `person`. Applications would be tasks like: *Select all court records, where people who have been born more than fifty miles from a given city of reference appear as interested party; select out of the running text all portions which are marked as «quotations»; compute a set of stylometric indices for the vocabulary used in such quotations.*

As already mentioned, we do not claim in this section, that our considerations here are general. We assume, that such models of the combination of structured representations of information and the covering text containing that information, can only be realized with data structures that allow inconsistent information to be handled and avoid normalization procedures.

As such models are few, we have so far only considered the problem of «hiding» κλειω data structures within the text. In the case of that system, the data are structured into a network which consists of entities[9]

called *groups* which among themselves can be linked by arbitrarily many relationships. Groups have an arbitrary number of attributes, called *elements*, which conceptually consist of variable length arrays allowing an arbitrary number of values, called *entries*, for each attribute.

To hide a network like that within a collection of information strings, as we described them so far, we see two possibilities:

On the one hand, the structural network of a data base could simply exist parallely to a text as we described it here, all entries of the network consisting of pointers to the relevant portion of the text, together with length information. (Obviously these pointers would have to consist of the nearest entrance point of the text plus an offset to the relevant portion.) The advantage of such a construction is obvious: existing database software could be taken over more or less unchanged and all textual functions needed, are contained per definition in a text engine, as we defined it so far. In our opinion this otherwise obvious model has, however, two major shortcomings: obviously updating the text could easily corrupt all the addresses contained in the entries of the data base. serious, as this problem is, we could handle it conceptually, by allowing for ↑DB links with a slightly modified definition, which would have to be inserted into the text at each point, which is the target of a data base referencing it. While this update problem is certainly tricky, it could eventually be mastered by the techniques hinted at.

Conceptually much more severe is a problem, which is introduced, when we consider even the most simple concept of data types. We already have introduced in section 3.1.3.4 above the concept of a *knowledge pointer* (↓KB). We did so, because of the necessity to treat «special kinds of text» in a special way. It is not sensible to sort calendar dates in alphanumeric fashion; and converting temporal notation related to the medieval saints into a notation that can meaningfully be sorted is no completely trivial task. This, however, is a typical data base problem. Of course a number of solutions could exist: one would be to include the

9. We discuss here the problem of hiding data structures within a general text representation. κλειω's groups are not entities in the sense of conventional DBMSs: to clarify, that they are not, we call them groups; as a precise definition of the differences would go beyond the scope of this paper, we do not give it however, but mention that envisaging them as entities will not be totally wrong. The same is the case for «elements» and «entries» introduced below.

functional equivalence of a [↓KB] into the data dictionary for this field – which means a type of redundancy which is dangerous. Another solution would be, to let the [↓KB] point to the data dictionary, rather than to the relevant knowledge bases to begin with. This could mean, that we need different implementations for a [↓KB] that occurs in a text without a hidden structured data base than the one used in texts with such an underlying structure; a bad solution. Avoiding that, however would bring us into the situation, where texts without an underlying dummy structure would not be allowed; not a bad solution, but an outright impossible one.

To avoid these situations, we propose therefore to discuss data models, which are per definition built into «natural texts». This would seem unusual from the point of view of other disciplines, but in history – or rather in the source-oriented model of historical dataprocessing – data bases, as this author has pointed out elsewhere, are always not so much collections of information, which define their own reality, but an attempt to structure information that has been tradited in a coherent form, i.e., usually as a text.

In the case of κλειω, we consider this possible, by allowing the text engine to administer three additional classes of constitutents of a string of information: *group pointers*, *element pointers* and *entry pointers* (denoted as [*Group*:Name∞] , [*Element*:Name∞] and [*Entry*:Name∞], respectively). They are defined as a generalization of the concept of a text reference introduced in section 3.1.3.2. The ∞ symbol replacing the ⇒ / ⇐ component of the [Name ⇒] / [⇐ Name] notation is assumed to indicate, that this generalization allows for an unlimited number of references starting from each embedded structural symbol, while our textual references provided always for precisely on physical reference.

2.2.2 The Interpretative Environment

The interpretative environment, which we have known intuitively so far as a kind of refinement to the concept of a printer driver consists of two independent components.

There exists a table which describes for each information string to be processed all the text qualities which it has, when any basic item is being encountered by any component of the text engine. This part of the environment has to be complete: that is the reason, why we introduced the concept of an entrance point into an information string. This part of the

interpretative environment is loaded whensosever a particular information string is presented to the text engine.

The second part of the interpretative environment describes not an information string, but the status of the text engine itself: it consists of applicability information for each possible textual quality and optionally a mapping of that quality to an input convention or an output property. This concept can best be interpreted, if we look at the idea of case sensitivity in text operations. Case sensitivity would be modeled in our concept as a state of the interpretative environment, where case is an applicable mode (and characters are during output mapped to different representations).

2.2.2.1 Applicability of Modes

The applicability of a mode consists of a statement, if the status of this mode shall be checked, when a basic item is being encountered by the text engine. If during printing the mode `german` is applicable, a mapping of characters with this mode into another font (or underlining) will be used; if it is not applicable, such mapping will be ignored. *Mutatis mutandis* this is also the case, when comparsions are performed.

2.2.2.2 Applicability of Style

The applicability of style consists of a statement, if it shall be checked when a basic item is being encountered by the text engine. For applicability style could be interpreted like a mode: if it is applicable, any difference in style will make two basic items unequal; there is no way to define a degree of difference.

2.2.2.3 Applicability of Color

The applicability of color defines whether optional I/O mappings shall be used, and a range, within which otherwise identical basic items have to be considered equal with regard to color. This range can be given as a pair of absolute numeric values, or as a table which specifies for each `color` in one information string, which `colors` in the other are acceptable.

2.2.2.4 Applicability of Size

The applicability of size defines whether optional I/O mappings shall be used, and a range, within which otherwise identical basic items have to be considered equal with regard to size. This range can be given as a

pair of absolute numeric values, or as percentage of the larger of two basic items, within which a smaller one is still to be considered equal.

2.2.2.5 Applicability of Views

As views are a more general form of the other textual qualities, their applicability is identical to that of the default modes of the four classes given above. The implementation considerations given in 3.1.2.5 define the difference between the applicability of any named view of the four types and the four default views of each type.

2.2.3 The Text Engine

What kind of activities our hypothetical text engine is performing, was shown intuitively more or less during the preceding sections. Obviously it is related to the production of output; obviously it is also performing comparison operations: our example of a «german sensitive» comparison as an equivalent to the optional «case sensitivity» of current text processing software implied so much. Though the definition of such a text engine is certainly not part of a discussion of text representation, we would like to include a brief, but somewhat more systematic, definition than given so far. The reason for this is, that we assume, that recent discussions on text representation had a tendency to concentrate a bit to much on printing. We would therefore like to describe operations, we think necessary to make use of all the qualities described in a more general way, i.e., influencing *any* kind of operation the type of text we describe here has to undergo.

Whensoever we define in the following sections an ability the «text engine shall have», this is a abbreviated expression therefore for the following reasoning: «We assume that processing historical data of requires a specific ability. If historical data are to be processed by more than one software system, we therefore need a standard for the encoding of the property calling for this ability.»

2.2.3.1 Texts Handled

The text engine shall be able to handle texts, which are mixtures of byte coded and bit mapped items. All items a text consists of has to be processable by all components in question. Which class an item has, has to be transparent for any applications programmer using tools provided by the text engine.

2.2.3.2 Import/Export

The text engine shall provide tools to import and export strings. This means, it shall be able to convert its own internal representation of an information string into a form that clearly distinguishes between different classes of items, specifically between byte coded and bit mapped ones, so software components, which do not have the ability to handle both, can extract those portions of a text, which they can handle. This export format, which is described as *external text format* has to be transferable on standard communication links.

2.2.3.3 Comparison and Sorting

The text engine shall have as part of its interface functions which are able to compare and sort sets of information strings, fully controlled by an interpretative environment as described above.

2.2.3.4 Searching

The text engine shall be able to use any informations string, which specifically includes such as contain bit mapped items, as a search key in the administration of large string collections, like dictionaries.

2.2.3.5 I/O

The text engine shall be able to convert its internal representation into a form which represents its various classes of items also on output devices, which do not provide means for the most obvious kind of presentation. It also shall contain parsing functions, which convert formats provided by input devices or software supporting sophisticated forms of input into a common internal presentation.

INDICE

AVVERTENZA

Nella collezione dei «**Contributi del Centro Linceo Interdisciplinare di Scienze Matematiche e loro Applicazioni**» *sono apparse le seguenti pubblicazioni*:

1. AGENO M., *Punti di contatto tra fisica e biologia* (con una Prefazione di Beniamino Segre. Corso di dieci lezioni tenute dal 22 al 26 maggio 1972), 1974.
2. ROSSI B., *Astronomia in raggi X* (Lezioni tenute nel febbraio e marzo 1972, raccolte da Bianca Maria Belli), 1974.
3. TOUSCHEK B., *Sull'insegnamento della teoria dei quanti* (Lezioni tenute nell'aprile 1972), 1975.
4. DIRAC P.A.M., *The Development of Quantum Mechanics* (Conferenza tenuta il 14 aprile 1972), 1974.
5. FERRARO V.C.A., *Il vento solare ed il campo magnetico interplanetario* (Conferenza tenuta il 17 aprile 1972), 1974.
6. *Seminari su*: «*La Scienza dei Sistemi*» (con una Prefazione di Beniamino Segre). Parte Prima (I Seminario: 30 novembre-4 dicembre 1970; II Seminario: 11-15 gennaio 1971), 1975.
   Parte Seconda (III Seminario: 8-12 marzo 1971; IV Seminario: 5-9 aprile 1971; V Seminario: 3-7 maggio 1971; VI Seminario: 24-28 maggio 1971), 1975.
7. *Seminario sulla*: «*Evoluzione Biologica*» (Roma, 10-11 gennaio, 17-19 aprile 1974), 1975.
8. NE'EMAN Y., *Patterns and Symmetry in the Structure of Matter* (Conferenza tenuta il 15 dicembre 1973), 1975.
9. SEIDENBERG A., *Constructions in Algebra* (Riassunto delle lezioni tenute nell'ottobre e novembre 1972), 1975.
10. *Tavola rotonda sul tema*: «*Problemi matematici ed economici odierni sulle assicurazioni*» (Roma, 24-25 novembre 1972), 1975.
11. CAMPA R., *La guerra e il processo di trasformazione tecnologica* (Conferenza tenuta il 26 maggio 1975), 1975.
12. MEDICI M., *Indirizzi verso motori automobilistici meno inquinanti* (Conferenze tenute nel marzo 1973), 1975.
13. *Colloquio sul tema*: «*Le tecniche di classificazione e loro applicazione linguistica*» (Firenze, 13 dicembre 1972), 1975.
14. GATTO R.R., *Interazioni elettromagnetiche, invarianza di scala e sue possibili estensioni* (Lezioni tenute nel settembre 1972), 1976.
15. *II Seminario sulla*: «*Evoluzione Biologica*» (Roma, 19-22 febbraio 1975), 1976.
16. DE GIORGI E., *Convergenza in energia di operatori ellittici* (Conferenza tenuta nel febbraio 1974), 1976.
17. MOISIL G.C., *Sur l'emploi des Mathématiques dans les Sciences de l'homme* (Conferenza tenuta il 5 giugno 1972), 1976.
18. ANDREOTTI A., *Lewy Problem for Cauchy-Riemann Equations* (Lezioni tenute nel febbraio 1973), 1976.
19. ALFONSI D., BALLA M.I., DE SANTIS F., GIORGI G., SCHAERF M., *Struttura di un sistema informativo per un servizio di documentazione scientifica* (Da una manifestazione tenuta nel febbraio 1976 per iniziativa del Centro Linceo e dell'Università di Roma), 1976.
20. TRUESDELL C.A., *Termodinamica razionale* (Corso di lezioni tenute nel gennaio 1973), 1976.

21. TOGNOLI A., *Introduzione alla teoria degli spazi analitici reali* (Lezioni tenute nel febbraio 1973, raccolte da Dina Smit Ghinelli), 1976.
22. HANSON A., REGGE T., TEITELBOIM C., *Constrained Hamiltonian Systems* (Ciclo di lezioni tenute dal 29 aprile al 7 maggio 1974), 1976.
23. CHESTNUT H., *Influence of Technology on Modern World Evolution and Use of Dynamic Models of Macro-Economic Systems in Development Planning* (Conferenza tenuta il 21 novembre 1972), 1976.
24. ANDREOTTI A., *Introduzione all'analisi complessa* (Lezioni tenute nel febbraio 1972), 1976.
25. REGGE T., RASETTI M., *Vortices and Current Algebra* (Conferenze tenute nel giugno 1975), 1976.
26. SANSONE G., *Studi sulle equazioni differenziali ordinarie nell'ultimo cinquantennio* (Lezione tenuta il 12 dicembre 1975), 1976.
27. SEGRÈ E., *Personaggi e scoperte nella Fisica contemporanea* (Ciclo di lezioni tenute dal novembre 1972 fino al marzo 1973), 1976.
28. *Seminario sui*: «*Sistemi di reperimento e selezione automatica dell'informazione*» (Roma, 17-21 aprile 1972), 1976.
29. *Seminario sulle*: «*Applicazioni della Scienza dei Sistemi alla Medicina e alla Chirurgia*» (Roma, 22-26 maggio 1972), 1976.
30. *Convegno Internazionale sul tema*: «*Trends in the Physics and Engineering of Technological Materials*» (Roma, 17-19 ottobre 1973), 1976.
31. *Gruppo di studio sui*: «*Fenomeni di alta energia nelle ultime fasi dell'evoluzione stellare*» (Roma-Frascati, 29 maggio-16 giugno 1972), 1976.
32. *III Seminario sulla*: «*Evoluzione Biologica. Il codice genetico*» (Roma, 26-28 febbraio 1976), 1977.
33. *Seminario sul tema*: «*Una nuova via italiana alla fisica delle alte energie: Ada, Adone ...* » (Roma, 24-25 maggio 1974), 1977.
34. DIRAC G., *Cardinal-determining Subgraphs of Infinite Graphs* (Lezione tenuta il 16 aprile 1975), 1977.
35. LEWY H., *On the Boundary Behavior of Holomorphic Mappings* (Lezione tenuta il 3 maggio 1976), 1977.
36. DUBOS R., *The Resilience of Ecosystems* (Lezione tenuta il 17 dicembre 1976), 1977.
37. *Seminario sul tema*: «*Rapporti tra Biologia e Statistica*» (Roma, 19-20 dicembre 1975), 1977.
38. BAER R., *Finite Metanilpotent Groups and Finite Sylow Tower Groups* (Corso di lezioni tenute nell'aprile 1976), 1977.
39. CESARI L., *Nonlinear Analysis and Alternative Methods* (Ciclo di lezioni tenute nell'aprile 1974), 1977.
40. *Convegno Internazionale*: «*Problemi connessi con l'utilizzazione pacifica dell'energia nucleare in Italia*» (Roma, 12-14 aprile 1976), 1977.
41. *IV Seminario sulla*: «*Evoluzione biologica*» (Roma, 17-19 febbraio 1977), 1978.
42. ISTRATESCU V.I., *Topics in Linear Operator Theory* (Corso di lezioni tenute nell'aprile 1976), 1978.
43. *Convegno sul tema*: «*Applicazioni del teorema del punto fisso all'analisi economica*» (Roma, 9-11 marzo 1977), 1978.
44. *Congresso Internazionale su*: «*L'insegnamento integrato delle Scienze nella scuola primaria*» (Roma, 7-15 gennaio 1976), 1979.
45. MARTINELLI E., *Introduzione alla teoria delle classi caratteristiche: uno sguardo panoramico* (Corso di lezioni tenute nel febbraio e marzo 1978. Redatte da Guido Lupacciolu e Paolo Piccinni), 1979.
46. ANGELINI A.M., *Linee di sviluppo nella utilizzazione della energia solare* (Conferenza tenuta il 9 marzo 1979), 1979.
47. BIETTI A., *Modelli matematici e statistici applicati all'Archeologia e alla Paletnologia* (Conferenza tenuta il 16 giugno 1978), 1979.
48. *V Seminario sulla*: «*Evoluzione Biologica. Evoluzione della sessualità ed evoluzione*

umana» (Roma, 23-25 febbraio 1978), 1979.

49. GOLINI A., *Attuali tendenze della popolazione in Italia: problemi e prospettive* (Conferenza tenuta il 9 febbraio 1979), 1979.
50. DE BENEDETTI S., *Dall'universo di Newton a quello di Einstein* (Conferenza tenuta il 28 maggio 1979), 1979.
51. *VI Seminario sulla*: «*Evoluzione Biologica. Ecologia ed etologia*» (Roma, 22-24 febbraio 1979), 1980.
52. LAUGWITZ D., *The Theory of Infinitesimals. An Introduction to Nonstandard Analysis* (Ciclo di lezioni tenute nel marzo 1977), 1980.
53. *International Meeting on*: «*Astrophysics and Elementary Particles, Common Problems*» (Rome, 21st-23rd February 1980), 1980.
54. CARERI G., *Ordine e disordine nella materia. Tre lezioni sugli aspetti interdisciplinari* (Roma, 26, 28 e 30 novembre 1979), 1981.
55. ADKINS W.A., ANDREOTTI A., LEAHY J.V., *Weakly Normal Complex Spaces*, 1981.
56. SAPORETTI C., *Risultati e prospettive dell'analisi dei testi accadici mediante il calcolatore elettronico* (Conferenza tenuta il 14 dicembre 1979), 1981.
57. *VII Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia*» (Roma, 28-29 febbraio - 1 marzo 1980), 1981.
58. RANZI S., *L'embriologia: recenti studi a livello molecolare* (Conferenza tenuta il 7 marzo 1980), 1981.
59. *Convegno sul tema*: «*Problemi di popolazione: realtà attuali e prospettive*» (Roma, 13 giugno 1980), 1981.
60. *VIII Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia. Aspetti biologici e sociali: parassitismo e simbiosi*» (Roma, 25-27 febbraio 1981), 1982.
61. BIETTI A., *Tecniche matematiche nell'analisi dei dati archeologici* (Ciclo di tre conferenze tenuto nel dicembre 1980), 1982.
62. ORLANDI T., *La filologia al calcolatore. Nuove prospettive per la letteratura copta* (Conferenza tenuta il 12 marzo 1982), 1982.
63. DE LUCA A., *La teoria generale dei codici* (Conferenza tenuta il 12 febbraio 1982), 1982.
64. *IX Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia*» (Roma, 24-26 febbraio 1982), 1983.
65. *Convegno sul tema*: «*Il miglioramento genetico dei cereali*» (Celebrazione del 40° anniversario della morte di Nazareno Strampelli - Roma, 10 dicembre 1982), 1983.
66. MOSCATI P., *Ricerche matematico-statistiche sugli specchi etruschi*, 1984.
67. MARTINELLI E., *Introduzione elementare alla teoria delle funzioni di variabili complesse con particolare riguardo alle rappresentazioni integrali*, 1984.
68. *X Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia. L'addomesticazione degli animali e delle piante*» (Roma, 24-26 febbraio 1983), 1984.
69. *Giornata di Studio sul tema*: «*Archeometria. Scienze esatte per lo studio dei Beni Culturali*» (Roma, 31 maggio 1983), 1985.
70. ARIAS P.E., DI BARI V.C., ORSOLINI RONZITTI G., *La ceramica attica a figure nere e rosse del Corpus Vasorum Antiquorum. L'analisi computerizzata dei dati*, 1985.
71. *XI Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia. L'evoluzione del comportamento e del sistema nervoso*» (Roma, 1-3 marzo 1984), 1985.
72. *Giornate di studio introduttive ai Seminari sulla «Scienza e Ingegneria dei Sistemi nelle sue più rilevanti applicazioni»* (Roma, 3-4 maggio 1983), 1985.
73. *XII Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia. Lo svolgimento della Genetica e dell'Evoluzione dopo la riscoperta delle leggi di Mendel*» (Roma, 13-15 febbraio 1985), 1986.
74. MOSCATI P., *Analisi statistiche multivariate sugli specchi etruschi*, 1986.
75. *Convegno sul tema*: «*Nuove frontiere dell'informatica: i sistemi esperti*», in collaborazione con la FINSIEL (Roma, 13-14 dicembre 1984), 1986.

76. *Meeting on*: «*Finite Thermoelasticity*» (Rome, 30th-31st May-1st June 1985), 1986.
77. *Seminario su*: «*La Scienza e l'Ingegneria dei Sistemi nella gestione delle acque*», in collaborazione con l'Istituto di Idraulica, Idrologia e Gestione delle Acque della Facoltà di Ingegneria dell'Università di Catania e con il FORMEZ (Roma, 20-22 novembre 1984), 1986.

Dal 1987 la collezione si chiama: «**Contributi del Centro Linceo Interdisciplinare Beniamino Segre**»

78. *XIII Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia. Evoluzione degli organuli cellulari*» (Roma, 26-28 febbraio 1986), 1987.
79. *XIV Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia. Dalla biologia dello sviluppo alle biotecnologie*» (Roma, 25-27 febbraio 1987), 1989.
80. *XV Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia. Le difese umorali e cellulari degli organismi*» (Roma, 24-26 febbraio 1988), 1990.
81. *Tavola rotonda sul tema*: «*Continui con memoria*» (Roma, 9 maggio 1986), 1990.
82. *XVI Seminario sulla*: «*Evoluzione Biologica e i grandi problemi della Biologia. La tettonica delle placche e la distribuzione dei viventi*» (Roma, 23-25 febbraio 1989), 1990.
83. *XVII Seminario sulla:* «*Evoluzione Biologica e i grandi problemi della Biologia. Origine ed evoluzione dell'uomo*» (Roma, 21-23 febbraio 1990), 1991.
84. *Conservazione del patrimonio culturale. Ricerche interdisciplinari*. I, 1992.
85. *XVIII Seminario sulla:* «*Evoluzione Biologica e i grandi problemi della Biologia. Sistematica ed evoluzione dei viventi* » (Roma, 26-28 febbraio 1991), 1992.

Per ordini d'acquisto rivolgersi a:
ACCADEMIA NAZIONALE DEI LINCEI
UFFICIO DIFFUSIONE PUBBLICAZIONI
Via della Lungara, 10
00165 ROMA
tel. (06) 683.88.31; telefax (06) 689.36.16