Cursul 12

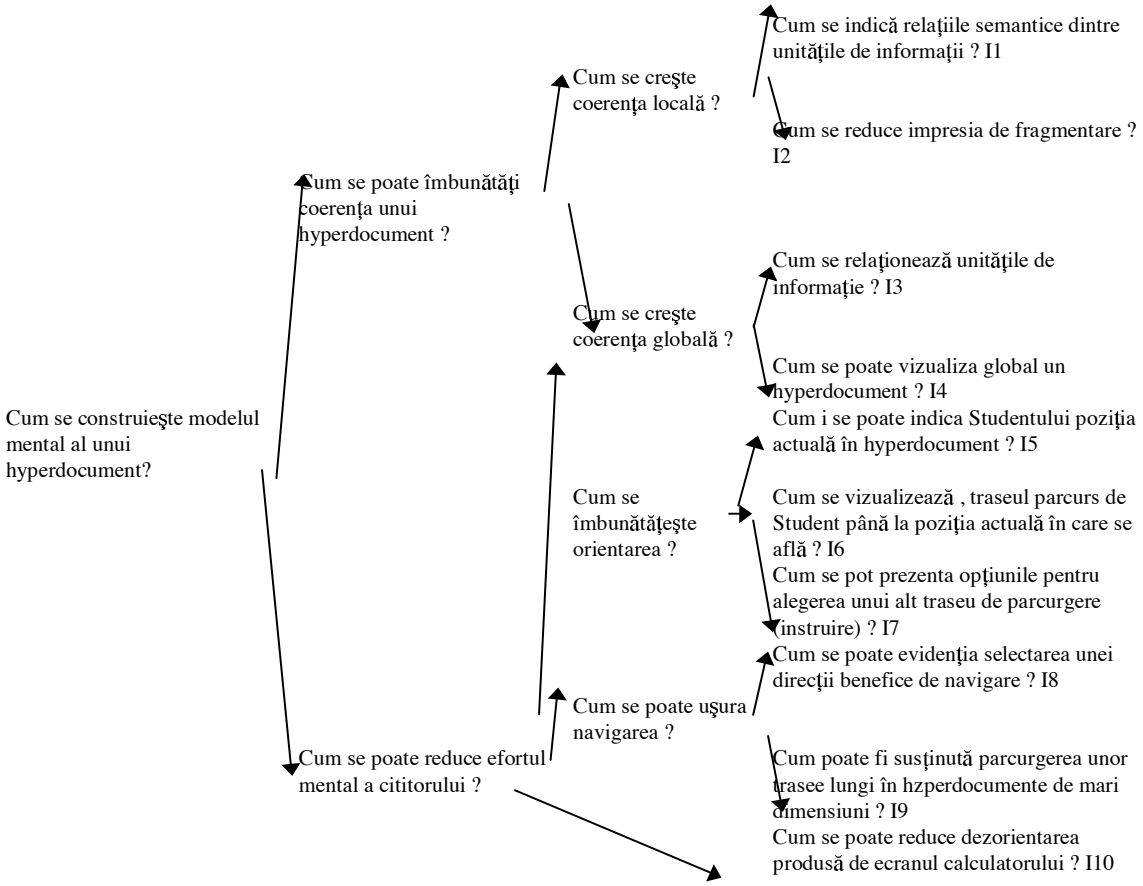

Figura 2. 18. Realizarea Modelului Mental al unui hyperdocument

Realizarea Modelului Mental al unui hyperdocument

Considerând opiniile cercetătorilor în domeniul Intermedia, ***cerinţele necesare integrării hypermed în mediul informatic*** (Figura 2. 18. Realizarea Modelului Mental al unui hyperdocument) su următoarele:

– Integrarea hypermedia în mediul informatic de tip desktop. Dispozitivul de Conexiuni (Link Engin trebuie să fie integrat în mediul informatic numai în modul în care este integrat astăzi sistemul de fişiere. U utilitar de înalt nivel sau o interfaţă pentru aplicaţiile de programare (API, application programmer interfac trebuie să fie oferită creatorilor de aplicaţii conţinând specificaţii pentru funcţionarea aplicaţiilor hypermedi

– Sistemele Hypermedia trebuie să ofere contexte multiple sau reţele multiple pentru a se putea exploata întregime legăturile de tip hypertext care conţin toate aplicaţiile.

– Aplicaţiile hypermedia trebuie să suporte filtre şi subansamble de explorare incrementale.

– Hypermedia Distribuite pe Reţele Mari (Wide Area Hypermedia) - Funcţionalitatea hypermedia trebuie fie extinse pentru a controla Reţele Mari (Wide Area Networks, WAN) nu numai Reţele Locale (LANs).

– Configurarea unui mediu integrat hypermedia este mult uşurată cu ajutorul tehnologiilor informati orientate obiect. De asemenea, cel mai logic Sistem de Management al Băncilor de Date (DBMS, Databa Management System) care poate fi utilizat pentru stocarea datelor (informaţiilor) cu privire la legătu (conexiuni) şi ancorări este Sistemul de Management al Băncilor de Date Orientat–Obiect (Object–Orient Data Base Management System, ODBMS).

Reguli de Proiectare pentru Documentele hypermedia

Principii		Unde sunt folosite Regulile de Proiectare
P1:	Se folosesc Conexiuni Etichetate , denumiri	I1+ şi I2
P2:	Se Indică similaritatea dintre unităţile de informaţii	I1 şi I2+
P3:	Se realizează concentrarea asupra contextului unităţii de informaţie	I1 şi I2+
P4:	Se floseşte un arc de curbă biunivoc între unităţile de informaţii	I3+
P5:	Se vizualizează structura documentului	I1, I2, I4+, I8 şi I9
P6:	Se folosesc cuvinte cheie (noţiuni) în structura de vizualizare în care este indicată poziţia actuală a Utilizatorului , traseul care l-a adus în acea poziţie şi opţiuni pentru trasee de continuare a navigării	I5+, I6+, I7+
P7:	Dezvoltarea unui set comprehensiv de mijloace de navigare care să poată vizualiza direcţiile şi distanţele	I8+ şi I9+
P8:	Folosirea unor diagrame cu ecrane de dimensiuni standardizate situate în pozii fixe	I10+

"+" indică importanţă prioritară, deosebită, specială

Figura 2. 19. Reguli de Proiectare pentru Documentele hypermedia

Pentru a oferi un mediu desktop total integrat având funcţionalitate hypermedia totală, Bieber propus un mecanism sistemic adecvat, amplu şi de mare capacitate bazat pe noţiunea de sisteme hypermed generalizate utilizând legile de conexiune (Bieber, 1993]. Acest mecanism poate interconecta în m independent aplicaţii de tip back-end ca de exemplu Sistemele Organice de Decizie, Sistemele Expe Băncile de Date şi front–end-urile (Aplicaţiile orientate de tip interfeţe ca de exemplu editoarele procesoarele de text, sau word processors, utilitarele de grafică) prin intermediul mecanismelor transmitere de comunicaţii tranzitorii. Legile de conexiune sistematizează corespunzător într-o diagran obiectele definite în back-end ca fiind modele, variabile, calculaţii atribuindu-le obiecte în front–end ca exemplu noduri, legături (conexiuni), şi conexiuni simbolice marcate (indicatoare).

Bieber a sugerat cerinţe caracteristice ale front–end şi back-end pentru strategia la nivelul sistem aplicată în vederea integrării hypermedia sau cooperării client / mecanism.

Caracteristicile Front–end-ului. Pentru ca mecanismul hypermedia să confere funcţionalităţi managementul legăturilor simbolice (marcate), comentarii, direcţii, diagrame concluzive, partiţii filtr navigări frontale şi trasee de întoarcere, front-end -ul trebuie să ofere următoarele operaţii caracteristi (Figura 2. 19. Reguli de Proiectare pentru Documentele hypermedia) :

Secvenţa de indicare a locaţiei obiectelor ca de exemplu marcajelor de conexiuni şi furnizar identificatorilor de locaţie mecanismului la selectarea unui marcator de legătură.

Front-end trebuie să solicite de la mecanism confirmarea permisiunii de editare pentru inser ştergeri, şi modificări.

Interfaţa Utilizator trebuie să furnizeze sistemului informatic hypermedia legături instantanee.

Când se salvează un document de pe ecran conţinând obiecte hypermedia împachetate, obiecte trebuiesc salvate de asemenea.

Caracteristicile Back-end -ului . Mecanismul hypermedia ar putea oferi funcţionalităţi ca exemplu: conexiuni (legături), adnotări, reacţii de răspuns privind traseele orientate, filtrare, şi conclu primite de la back-end. Back-end -ul oferă următoarele funcţionalităţi:

Oferă informaţie specifică despre structura and documentele aplicaţiilor sale.

Legile (regulile) de interconexiune trebuiesc descrise de realizatori. Acest lucru poate fi realizat ajutorul unui editor pentru legi de conexiune care să poată fi utilizat în locul unei scrieri logice complicate.

Back-end -urile pot oferi informaţii pentru control şi mecanisme de interpretare aferente obiectel care sunt transmise prin intermediul mesajelor de comunicaţii. De exemplu, obiectele care trebuie servească drept indicatori / marcatori de conexiuni relaţionale pot fi etichetate.

Back-end -urile pot realiza comenzi de control asupra mecanismului hypermedia ca şi front-end -uri (lista de comenzi şi informaţii sensibile la context).

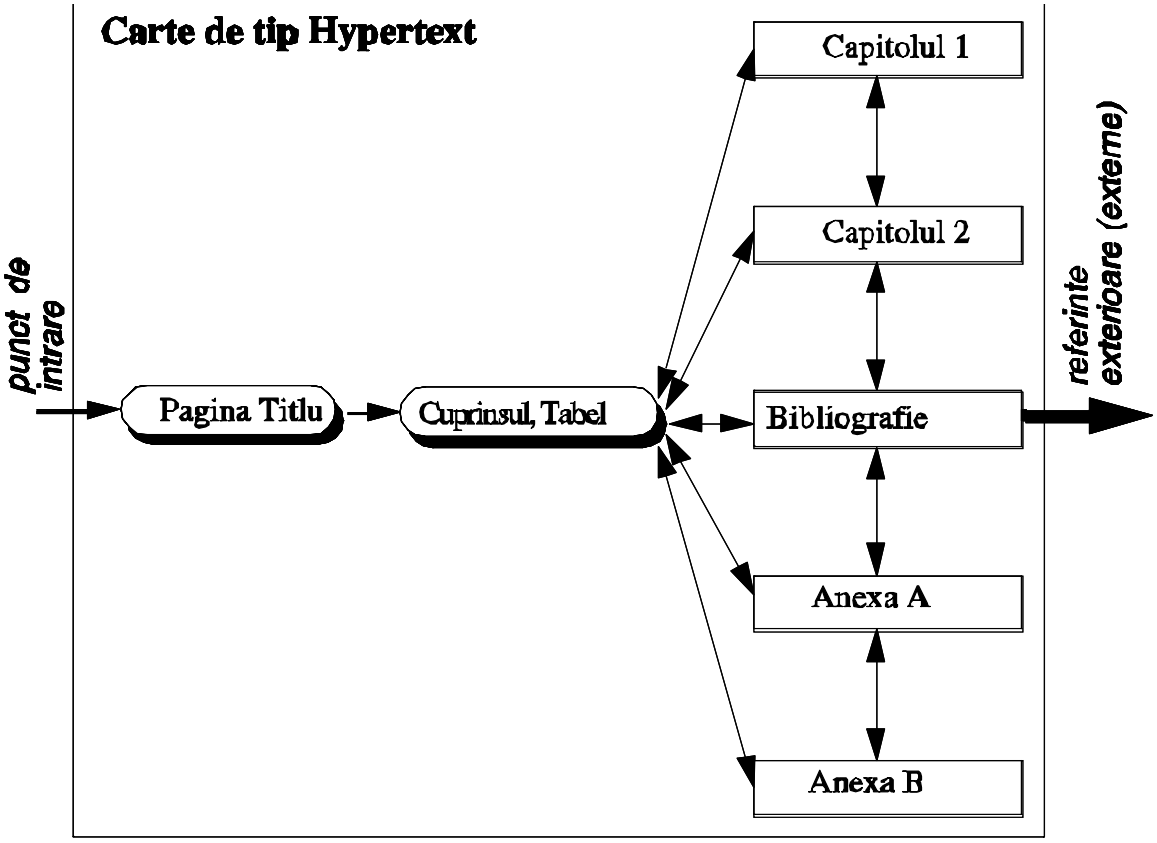

Figura 2. 20. Componentele unei cărți electronice de tip hypertext

Back-end trebuie să incorporeze un standard de transfer / comunicații pentru documentelor ca exemplu ODA sau SGML.

Într-un mod similar cu mecanismul de conexiuni Intermedia şi mecanismul hypermedia a lui Bieb serviciul de conexiuni al lui Sun oferă un protocol extensibil pentru creearea şi menținerea relațiilor dint aplicațiile autonome de tip front-end [Pearl, 1989]. În mod similar cu metodele prezentate anterior, editarea stocarea obiectelor informatice este administrată de către aplicațiile independente care de asemenea oferă u număr oarecare de operații de tip front-end asupra legăturilor. Serviciul de conexiuni stochează / arhivea numai reprezentarea nodurilor nu şi nodurile. Astfel, definirea şi granularitatea / mărimea nodurilor su cedate către aplicațiile individuale. De asemenea, stocarea informației în nod este independentă de stocar informației în conexiune.

Serviciul de Conexiuni (Link Service) facilitează dezvoltarea aplicațiilor de adăugare funcționalitații hypertextului cu ajutorul unui simplu protocol, un server utilizat în comun de back-end s serverul de conexiuni, o bibliotecă, şi utilitarele pentru administrarea băncii de conexiuni (A se vedea Figu 2. 20. Componentele unei cărți electronice de tip hypertext). Aplicațiile comunică cu serverul de conexiu prin intermediul protocolului de Servicii de Conexiuni (Link Service protocol). Acest serviciu permi aplicațiilor independente să integreze mecanismele de conexiuni in interiorul funcționalității standard prop şi astfel să devină parte integrantă a unui sistem hypertext deschis şi extensibil. Textul existent şi editoare grafice pot fi integrate într-o astfel de structură fără nici o modificare. Datorită separării dintre nod conexiunii informatice, Serviciul de Conexiuni nu asigură controlul specific, editoarele pentru conținut nodurilor, accesul multi-utilizator concurent analog paralel, sau alte forme de integrare a datelor.

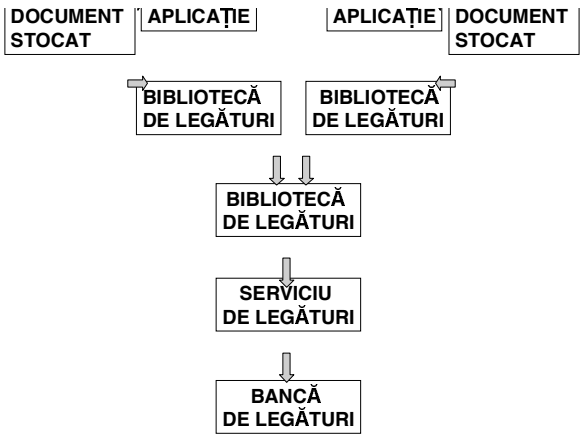

Figura 2. 21. Serviciul de Legături interconectate

O Arhitectură pentru Hypertextul uşor accesibil (Deschis). Câteva din elementele implicate dezvoltarea (realizarea) unui asemenea sistem hypertext deschis includ următoarele componente [Pea 1989]:

Interfața Utilizator pentru crearea şi administrarea (managementul) legăturilor care ar trebui să fie concordanță cu editoarele folosite în aplicațiile individuale diverse.

Dacă ***Serviciul de Conexiuni*** şi aplicațiile sunt decizii de procese distincte trebuie responsabilitatea de divizare / alocare / partiționare să aparțină acestui serviciu cu excepția strategiilor manipulare a informațiilor şi a dialogurilor cu utilizatorii.

Serviciul de Conexiuni / Legături (Figura 2. 21. Serviciul de Legături interconectate) poate detec şi suspenda / elimina conexiunile întrerupte, fie implicit fie explicit. Eliminarea implicită poate avea l atunci când un utilizator încearcă să acceseze o legătură / conexiune de la sfârşitul valid al acesteia căt sfârşitul invalid al acesteia recomandând utilizatorului să elimine conexiunea. Eliminarea explicită poate av loc, prin intermediul unui mecanism de colectare a refuzurilor de conectare, prin intermediul conexiunil direcționate, prin validarea nodurilor şi eliminarea conexiunilor / legăturilor invalide.

În timp ce interpretarea obiectelor informatice poate fi cedată aplicațiilor individuale, Serviciul Legături trebuie manipuleze autonom interpretarea legăturilor. Pe de altă parte, consistența interpretă hypertextului nu poate fi garantată dacă nodurile sunt interpretate în mod separat în raport cu legăturile.

Documentele nestructurate cum ar fi fişierele de tip **ASCII** nu pot fi manipulate elegant decât num în cazul în care sunt unic indexate sau dacă transmit o semantică precisă.

Problematica traversării conexiunilor în cadrul unei rețele şi localizarea obiectelor dispuse la ma distanță unul față de altul este foarte importantă datorită performanțelor care trebuiesc îndeplinite şi factoril de costuri. De asemenea, trebuiesc luate decizii asupra locului de plasare a Serviciului de Conexiuni în rețe O altă problemă corelată este apelarea unei aplicații care în mod obişnuit nu poate rula în momentul în ca utilizatorul operează cu o conexiune corespunzătoare unui nod administrat de respectiva aplicație.

Set de programe care creează Hypermedia (Hypermedia Toolkit) . Strategia toolkit menționa de Haan et al., a fost obținută de Puttress şi Guimaraes. Puttress şi Guimaraes au propus un toolkit care poa fi folosit de creatorii de aplicații pentru a adăuga funcționalități hypermedia toolkit-urilor deja existen aplicațiilor independente sau specifice sau mediilor informatice [Puttress & Guimaraes, 1990]. Arhitectu toolkit-ului hypermedia este similară altor arhitecturi de tip multi-conexiuni (Vezi Figura 3). Straturile sur Software pentru Aplicații, Nivelul Toolkit-ul Hypermedia, Sistemul de Stocare, şi Sistemul de Reprezentare ***Toolkit-ul hypermedia*** constă din următoarele trei componente:

a) ***Interfața Sistemului de Stocare*** (denumită şi Eggs): Această interfață constă într-un set de clase de t **C++**, care conferă o structură hypermedia aplicațiilor informatice stocate. Interfața Sistemului de Stoca furnizează structura de reprezentare grafică schematică dintre aplicația superioară şi sistemul de stoca subordonat. Astfel, sistemul de stocare poate fi modificat sau schimbat fără a modifica aplicația. În m similar structurii de tip HAM, modelul informatic este realizat cu ajutorul grafurilor, contextel condiționale, nodurilor, legăturilor / conexiunilor, atributelor, şi simbolurilor. Această interfață interpretează nodul informatic - Interfața Sistemului de Stocare este precis considerată ca flux de bytes fă structură sau mesaj /interpretare. Interfața Sistemului de Stocare asigură mecanismele de control ale versiur şi simultaneității / corespondenței. La acest nivel administrarea transferurilor / tranzacțiilor este delicată s controlul aplicației.

b) *Interfața Aplicației:* Această interfață este compusă din obiecte de tip date (informații) care comunică c aplicația mai sus menționată.

c) *Interfața Sistemului de Reprezentare:* Această interfață este responsabilă pentru prezentarea ideilor imaginilor utilizând mediul interfeței de tip utilizator, în mod independent de platforma de prezentar Interfața pentru Aplicație şi Interfața de Reprezentare sunt realizate utilizând un set de clase de tip C+ denumite împreună set de utilitare orientate obiect de tip hypermedia (Hypermedia Object-oriented Toolk HOT). HOT oferă abstractizările cerute pentru aplicațiile hypermedia în acelaşi timp incluzând (încapsulân şi detaliile necesare pentru stocare (arhivare) cât şi sistemele de reprezentare. HOT conține clase de da (informații) care includ: HGraph, HContext, HNode, and HLink. HOT conține de asemenea clase c vizualizare pentru fiecare din clasele de date: HGraphView, HContextView, HNodeView, HLinkView ar Hframe.

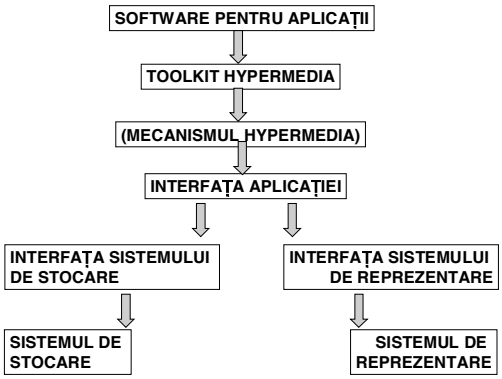

Figura 2. 22. Arhitectura Documentului Hypermedia

Această arhitectură (Figura 2. 22. Arhitectura Documentului Hypermedia) va fi extinsă pentru putea suporta medii multi-utilizator , pentru a oferi mijloace efective de distribuție şi comunicații înt utilizatorii aplicațiilor hypermedia, şi explorarea mijloacelor de dezvoltare a aplicațiilor hypermed colaborative.

Modelul de Proiectare Hypermedia (Hypermedia Design Model, HDM) este un model hyperte care a fost dezvoltat ca parte a proiectului HYTEA de către Consorțiul European [Garzotto et al., 199 Caracteristicile fundamentale ale HDM includ reprezentarea aplicațiilor hypertext cu ajutorul primitivelc entități tipice compuse din ierarhii de componente; perspective diferite pentru fiecare componentă; unit corespunzătoare perechilor componente de poziționare; cantități reprezentând conținutul actual de unită legături structurale asociind componentele aparținând aceleiaşi entități; legături pentru aplicații corelâ componente aparținând unor entități diferite; şi semantici contemplative / analitice determinând vizualizar şi proprietățile dinamice ale aplicațiilor. Aceste primitive sunt similare obiectelor definite în HAM.

Modelul Hypertext de Referință de tip Dexter (Figura 2. 23. Modelul de Referință de t Dexter) deține abstractizările importante ale caracteristice sistemelor hypertext existente şi viitoare [Hala & Schwartz, 1990]. Obiectivul acestui model este să ofere o bază sistematică pentru compararea sistemelor să dezvolte standarde pentru comunicații şi inter–operabilitate. Modelul Dexter divide un sistem hyperte system în trei nivele :

a. Nivel Runtime. Acest nivel administrează prezentarea hypertextului şi dinamica interacțiunilor c utilizatorul. Deoarece modelul de prezentare este prea complex şi divers pentru a fi prezentat în cadrul un model general, modelul Dexter nu oferă detaliile mecanismului de prezentare. Cu toate acestea, mecanisme de prezentare pot fi definite specific şi pot conține informații despre modul în care o componentă / rețea es prezentată utilizatorului. Aceste stipulații de prezentare constituie o interfață între nivelul runtime şi nivel de stocare.

b. Nivelul de Stocare . Nivelul de Stocare este trăsătura caracteristică principală a modelului Dexter. Nivel de Stocare modelează o bancă de date care este alcătuită dintr-o ierarhie de date-conținând componente ca sunt interconectate prin conexiuni relaționale. Componentele au identificatori unici şi conexiunile pot identificate cu ajutorul unui set sau mai multe seturi de identificatori de componente. Componente corespund noțiunii generale de noduri şi pot conține text, grafică, imagini, audio, video etc. Componente sunt tratate drept containere universale care conțin date şi modelul nu specifică în nici un fel structura de da aflată în interiorul containerelor. Astfel, nivelul de stocare nu face diferența dintre componentele de text componentele de grafică. Nivelul de Stocare se concentrează în principal asupra mecanismelor prin ca componentele şi conexiunile sunt legate împreună pentru a forma rețeaua hypertext.

c. Nivelul Interior Componentei . Nivelul Interior Componentei se ocupă de conţinutul şi structura d interiorul componentelor reţelei hypertext. Dacă extinderea posibilă a structurii / a conţinutului informat care poate fi inclus într-o componentă este nelimitată (open-ended), modelul Dexter tratează acest niv (strat) ca fiind exterior / în afara scopului propus de modelul Dexter. Ipoteza consideră că structu documentului este modelată în conformitate cu ODA, SGML, IGES etc., şi în consecinţă structu documentului va fi utilizată împreună cu acest model pentru a capta conţinutul / structura. Pe de altă parte, interfaţă critică între nivelul / stratul de stocare şi stratul din interiorul componentei numit ancorare se refe la mecanismul de adresare al locaţiilor sau obiectelor din interiorul conţinutului unei componen individuale. Ancorele pot fi identificate cu ajutorul unui identificator unic de fixare.

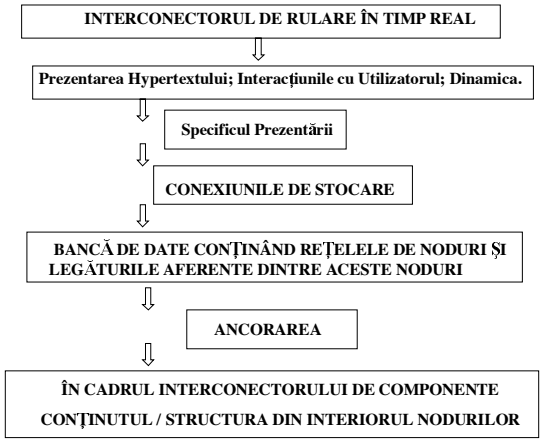

Figura 2. 23. Modelul de Referinţă de tip Dexter.

Halasz şi Schwartz afirmă că nu există suport de sistem pentru toate mecanismele componen menţionate de modelul Dexter. Cu toate acestea, sistemele existente sunt încă comparate cu acest mod Modelul a fost utilizat la dezvoltarea Formatului de Transfer de tip Dexter, care este un standard de transf pentru hypertext.

Modelul de Referinţă pentru Hypertext de tip Trellis sau "***modelul-r***" consideră hypertextul un ansamblu de diferite nivele de abstractizări [Furuta & Stotts, 1990]. În general, hypertextul poate fi diviz în :

a) Nivelul Abstract: Acest nivel / strat este definit abstract sub forma unor componente independente ca sunt conectate împreună într-o anumită metodă convenţională. Nivelul abstract nu descrie detaliile prezentare.

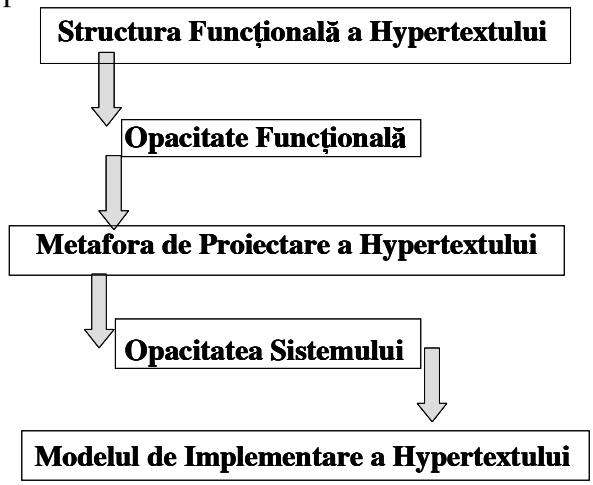

Figura 2. 24. Structura Generală caracteristică pentru Funcţionalitatea Hypertextului.

b) Nivelul Concret: Stabileşte reprezentările concrete în care caracteristicile de reprezentare grafică fizică hypertextului. Cu alte cuvinte, conţinutul fiecărei ferestre este specificat dar nu este şi sistematizat.

c) Nivelul Vizibil: Acest nivel este responsabil pentru interconectarea şi prezentarea reţelei hypertext pe u terminal fizic. Reprezentările în nivelul abstract sunt făcute la cel mai înalt nivel de abstractizare în timp reprezentările în nivelul vizibil sunt la un nivel scăzut. Deoarece nivelul abstract poate fi standardiz utilizând protocoalele de schimburi de documente ca de exemplu SGML, nivelul vizibil poate fi standardiz utilizându-se X Windows.