Hypermedia

Prof. dr. Paul De Bra
Prof. dr. Lynda Hardman
Eindhoven Univ. of Technology

Topics

- Hypermedia
  - What is hypermedia ?
  - Brief historical overview
  - Architecture of hypermedia systems

What is Hypermedia ?

- Hypermedia ≠ World Wide Web
  - The Web is one type of hypermedia application, but does not illustrate all hypermedia concepts.
- Hypermedia = (multimedia) information objects + navigation through hyperlinks.
  - *nodes*: information objects, displayed simultaneously or not, in one window or in separate windows
  - *links*: unidirectional or bidirectional, embedded in the information object or dynamically generated and possibly displayed on the side.

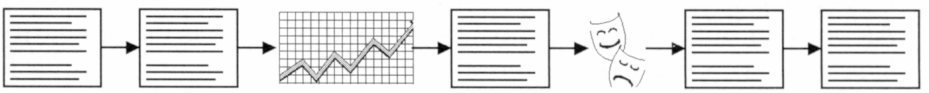
Definition

- There are many different definitions of hypermedia. Two examples:
  - Hypermedia is a database that has active cross-references and allows the reader to “jump” to other parts of the database as desired.
  - Hypermedia is an application which uses associative relationships among information contained within multiple media data for the purpose of facilitating access to, and manipulation of, the information encapsulated by the data.

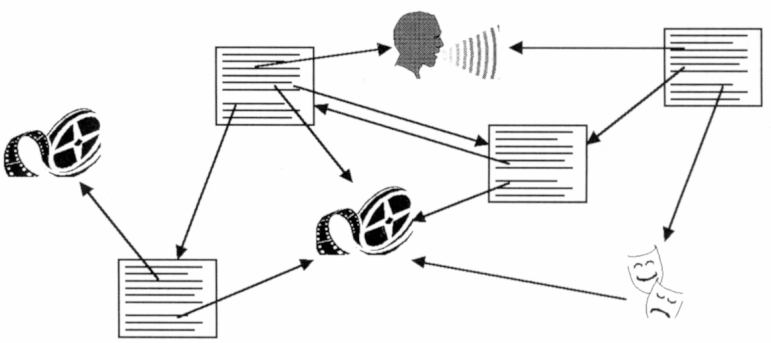
Alternative definitions

- Rao and Turoff (cognitive science, 1990): *It is appropriate to view hypertext as a method of supporting the expression of relationships among objects in a database. Hypertext should be treated as a general-purpose tool with approaches to handling nodes, links, and retrieval, that fits within the context of any application and convey common meanings to the users.*
- J.D. Bolter (literature, 1989): *A hypertext consists of topics and their connections, and [...] the topics can be paragraphs, sentences, or individual words. A hypertext is like a printed book that the author has himself attacked with a pair of scissors and cut into convenient verbal sizes. The difference is that the electronic hypertext does not simply dissolve into a disordered bundle of slips; the author defines its structure by establishing electronic connections among the slips.*

Hypermedia is non-linear

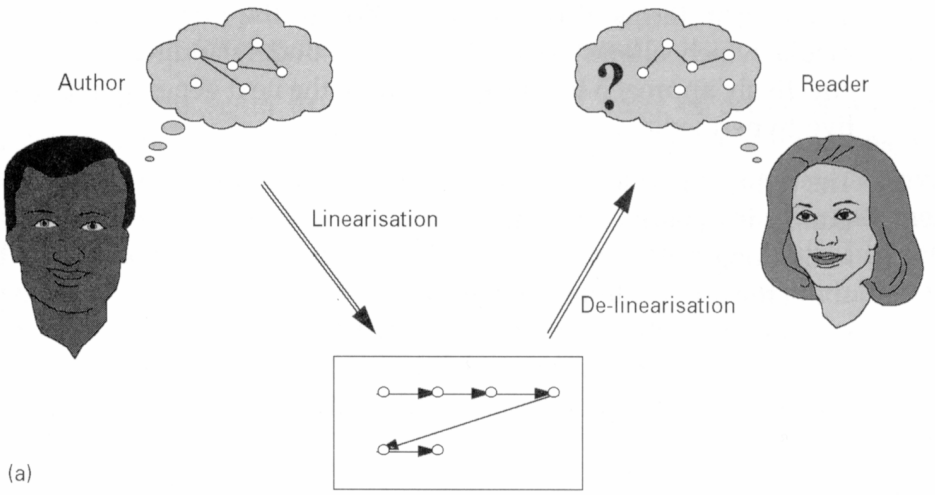


Linear 'linking' of traditional media

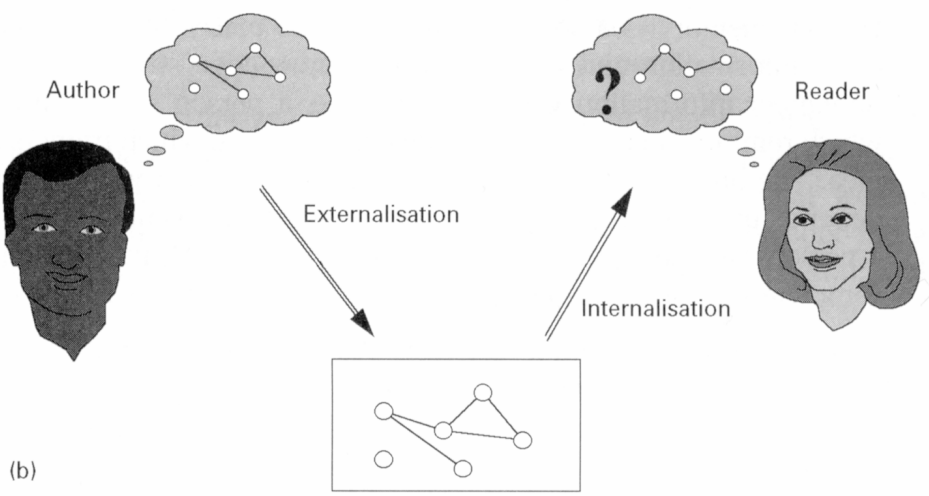


Nonlinear 'linking' of hypermedia

Problems with linear media



Solution through hypermedia



Hypermedia linking

- **Structural links**: used to provide organization in the information chunks, (e.g. navigation through a hierarchy of chapters, sections and paragraphs)
- **Associative links**: semantic relationship between different information elements (e.g. a cross-reference to related information)
- **Referential links**: between an item and an "elaboration" of the item, (e.g. an example or an explanation)

Hypermedia applications

- Wide variety of application areas:
  - **small size**: memos, letters, hypertext fiction, announcements, advertisements.
    completely authored manually.
  - **large size**: service manuals, technical documentation, educational material, archives, encyclopedia, libraries.
    information content often stored in databases.
    many links often generated automatically (e.g. based on occurrence of words or terms).

Problems with scalability

- Development cost for content may scale well. Cost increases less than linear with size.
- The link structure may become very complex. (Selecting good associative links becomes harder as the number of potential link destinations grows.)
- Very large hypermedia applications involve a lot of information that needs to be updated frequently. (E.g. an encyclopedia)

Hypermedia **engineering**

- Developing large hypermedia applications is like developing large information systems applications.
  - The same kind of project planning and management techniques can be used. (e.g. waterfall model)
  - There will be a close relationship between the desired hypermedia *product* and the development *process*.
  - Hypermedia requires a *fine grained maintenance*. (certain nodes must be updated frequently.)

Web-based hypermedia

- Additional issues for Web-based hypermedia applications include:
  - *Analysis of interconnectivity*: provide links to related material available on the Web.
  - *Bandwidth considerations*: facilitate the use through slow lines, e.g. provide thumbnail images.
  - *Maintenance*: links to other sites become stale and need to be validated frequently.
  - *Multi-platform*: verify that the application works with different browsers on different platforms.
  - *Multi-user support*: support many and also *different* users of the website (i.e. personalization)

History of hypermedia

- The ideas of hypermedia (information objects and associative linking) stem from Vannevar Bush, in 1945.
- The first hypermedia concepts and systems appeared between 1965 and 1970. (Ted Nelson, Andries van Dam, Doug Engelbart)
- Between 1975 and 1990 several influential systems were built, including NoteCards, Intermedia and HyperCard.
- Since 1990 the Web has dominated the hypermedia field.

1945: Vannevar Bush and Memex

- Vannevar Bush developed the concept of the "Memory Extender" Memex:
  - mechanized device using microfilm (small enough for office use);
  - documents can be annotated;
  - documents can be linked so that when one document is retrieved some other documents are automatically retrieved as well;
  - trails of documents (interesting reading sequences, with annotations) can be photographed and mailed to other users.

1965: Ted Nelson and Xanadu

- Xanadu aims at creating a unified literary environment on a global scale, a *repository* of *everything* that anybody has ever written.
  - Ted Nelson coined the term **hypertext**.
  - Xanadu is a client-server system, with a strong emphasis on the server side.
  - Xanadu offers *transclusion*, the inclusion of fragments of other documents in new documents.
  - Xanadu has a strong addressing mechanism to support transclusion. (similar to URLs in WWW)

1967: Hypertext Editing System

- Andries van Dam, Brown University.
  - First working hypertext system;
  - Ran on IBM/360 with 128K memory;
  - Used for documentation of the Apollo space program.
- In 1968 Andries van Dam developed FRESS, a *File Retrieval and Editing System*.
  - multi-user timesharing version;
  - commercially reimplemented by Philips;
  - rich hypermedia functionality.

1968: Augment / NLS

- Doug Engelbart started the Augment project in 1962 and completed NLS, the oN Line System in 1968.
  - Augment aims at augmenting the human capabilities and productivity through hypertext.
  - NLS was an experimental tool to store specifications, plans, designs, programs, documentation, reports, etc.; it grew to over 100.000 information items.
  - Doug Engelbart now heads the Bootstrap Institute (www.bootstrap.org).

1975: ZOG, and later KMS

- Donald McCracken and Robert Akscyn developed ZOG at CMU, and later founded *Knowledge Systems* and created KMS.
  - ZOG used “frames” with a standard layout, for use on text-based terminals; links were not embedded in the text.
  - ZOG ran on PERQ workstations, in a multi-user environment;
  - ZOG was used as an information management system for the USS Carl Vinson. (This ship had 28 workstations on board).

Screen shot: KMS

1978: The Aspen Movie Map

- Developed by Andrew Lippman at MIT.
  - First true **multimedia** system: used videodisks to offer a virtual tour through Aspen;
  - One screen showed the video, another screen showed the map of Aspen;
  - The user could “drive” around Aspen and enter certain buildings.
  - High performance: 30 images per second possible, but normally slowed down to 10.
  - Proof of concept, later applied to the *Movie Manual*, for technical maintenance and repair.

The Symbolics Document Examiner

- In 1985 Janet Walker completed the SDE project (started in 1982) to make all documentation for the Symbolics computer electronically available through hypertext.
  - 10.000 node hyperdocument;
  - 23.000 links;
  - 10 Mbyte of (text) storage;
  - written using a special authoring interface: *Concordia*;
  - bookmarking facility for users.

NoteCards

- Also in 1985, Frank Halasz developed NoteCards at Xerox Parc.
  - each node can contain an arbitrary amount of information;
  - over 50 types of nodes, including different types of overviews;
  - links are *typed* connections between nodes;
  - link destination can be shown in a separate window; the user can open arbitrarily many windows.

Intermedia

- Developed at Brown Univ. in 1985, Intermedia was the most sophisticated hypermedia system of its time.
  - ran on Macintosh with Unix (A/UX);
  - linking protocol allows other applications to be integrated into Intermedia hyperdocuments;
  - overview nodes offer authored overviews of parts of the hyperdocument;
  - *web views* offer dynamically constructed graphical overviews of the link structure.

1986: Guide

- Developed by Peter Brown around 1982, and available on Macintosh around 1986.
  - commercialized by OWL, and first hypertext system available for Mac and IBM-PC.
  - main linking mechanism is the *replacement* link: node breaks open at the position of the link and the link anchor is replaced by a piece of text;
  - annotations are possible through pop-ups;
  - *jumps* represent “normal” hypertext links;

1986: Hyperties

- Developed by Shneiderman, and available on the IBM-PC.
  - first hypertext system for the IBM-PC
  - links are tied to words or phrases;
  - for each word or phrase only one link destination is possible; (this is somewhat limiting but offers stable link destinations)
  - small and fast; uses only a text window and no mouse.

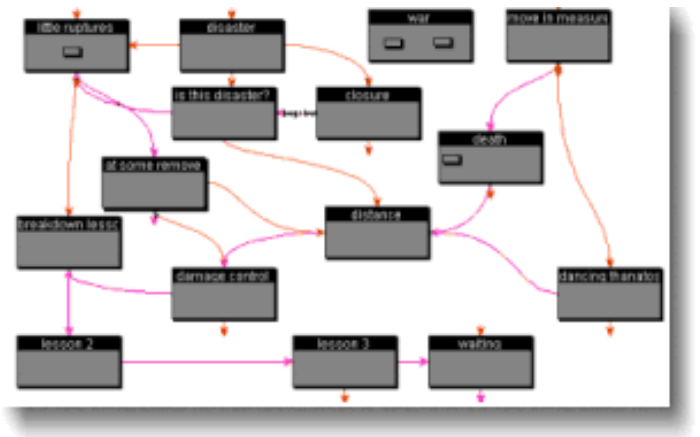
1987: HyperCard

- Apple bundled HyperCard (for free) with every Macintosh since 1987.
  - uses stacks of cards, and links between cards (in the same or different stacks);
  - powerful scripting language lets authors develop prototype user-interfaces (and not just hypertext systems);
  - creating hypertext using HyperCard is more like *programming* than like *authoring*.

StorySpace

- Developed by Mark Bernstein and his company *Eastgate Systems*.
  - used for hypertext fiction, such as Michael Joyce's "Afternoon";
  - has a powerful scripting language;
  - has several ways to produce overviews;
  - available for Macintosh and Windows;
  - basically the only commercial hypertext system that survived the advent of World Wide Web.

StorySpace



World Wide Web

- Basic architecture and first browser and server developed by Tim Berners Lee at CERN, in 1990.
- Mosaic browser for X-Windows developed by Marc Andreessen in 1992.
- Architecture vaguely resembles Xanadu.
- Developments are governed by the World Wide Web Consortium (W3C).

Goals of Hypermedia

- Support actions which result in the identification of appropriate information.
  - finding through (associative) links;
  - recognizing the meaning of links (or link labels);
  - search tools, based on user profiles.
- Support actions which facilitate the effective use of information.
- Support actions which result in control of appropriate information.

(Lowe and Hall)

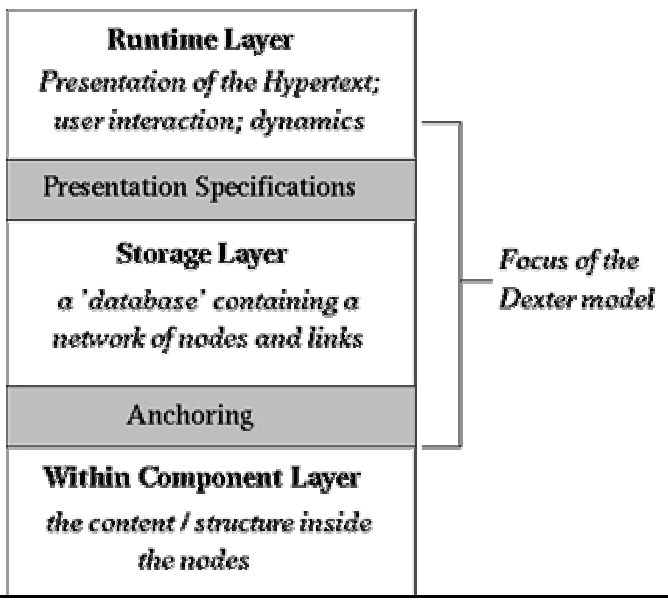
Data, Information, Knowledge

- **Data**: artifacts which exist as a vehicle for conveying information;
- **Information**: the interpretation of data within a context set by *a priori* knowledge and the current environment;
- **Knowledge**: the base of personal information which is integrated in a fashion which allows it to be used in further interpretation and analysis of data.

Hypermedia System Architecture

- General requirements for hypermedia systems:
  - *Composite components*: multiple media items must be combined into a single presentation, and presentations into larger units;
  - *Dynamic data*: certain media items involve timing constraints and temporal relations between items;
  - *Links as objects*: links need to have a "context" and attributes; they are more than just links;
  - *Separation of data, information, application and presentation*;
  - *Openness*: allow importing new objects.

The Dexter Model



Dexter: the storage layer

- a hypertext consists of *components*, with a *resolver* and *accessor* function.
  - components can be atoms, links or composite entities;
  - every component has a globally unique identity;
  - links can be uni- or bidirectional or undirected, between two or more components;
  - components can be found based on properties, through the *resolver* function;
  - components can be accessed when their id is given, through the *accessor* function.

Dexter: Anchoring

- Anchors (with an id and value) link the abstract components of the Storage Layer with actual components in the Within-Component Layer.
  - The anchor value may indicate a location, region, item or substructure within a component.
  - The anchor value may change because of dynamic behavior of the within-component layer, but the anchor id remains the same.
  - The within-component layer is a black box.

Dexter: Presentation Specifications

- Endpoints of links are defined by *specifiers*, which consist of a component specification, and anchor id, a direction and a presentation specification.
  - The specifier specifies which anchor in which component a link connects to.
  - An *instantiator* function specifies how the component is to be "presented" by the system; it forms the link to the Runtime Layer.

Dexter: Runtime Layer

- Takes care of *instantiating* component.
  - instantiating a component is a result of instantiating an anchor;
  - the runtime layer includes *sessions*, to keep track of the moment-by-moment mapping between components and their instantiation;
  - a runtime *resolver* function can use information from the session to retrieve components;
  - a component can also be *realized* which is done to save editing operations.

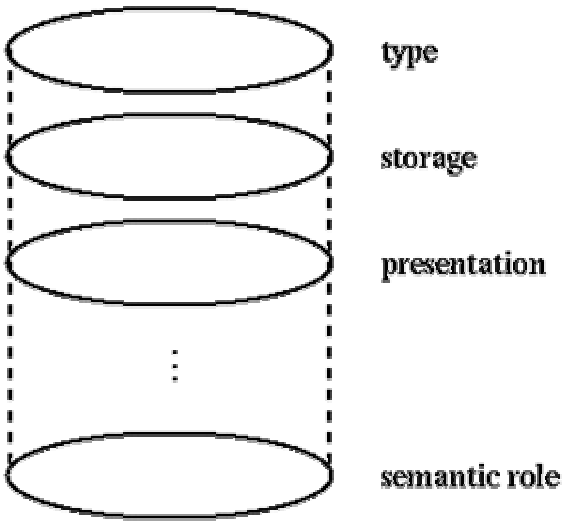
The Tower Model

- The Tower Model is completely object oriented. It distinguishes:
  - *nodes*: basic information objects
  - *links*: objects with source and destination anchors
  - *anchors*: objects with value indicating where the anchor is tied to a node
  - *composites*: both nodes and links can be composite
  - *tower objects*: represent aspects of objects
  - *city objects*: represent views of objects

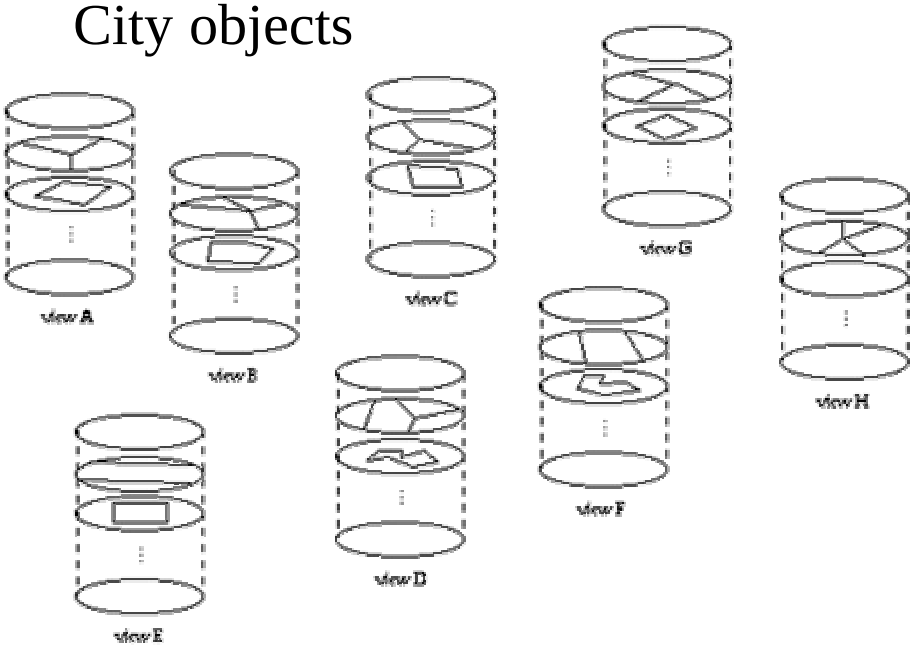
Basic/Composite objects

- Basic nodes, links and anchors are black boxes and not described within the model.
- Composite Objects consist of:
  - a *kind*: node, link or anchor;
  - a *set of components* (nodes, links and anchors) making up the composite;
  - a *structure constructor* for the composite;
  - a set of *locations* of the components;
  - a set of *global constraints*;
  - some *global information*.

Tower objects



City objects

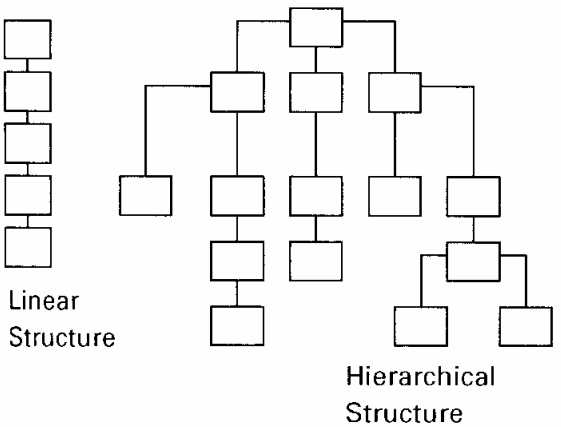


Hypermedia design

- Hypermedia design influenced by:
  - information retrieval and accessibility;
  - information security (make sure only the right people can access the right information);
  - information reuse and maintainability;
  - inter-relationships between information sources (creating the right links);
  - provision of differing viewpoints on the information (i.e. personalization and adaptation)

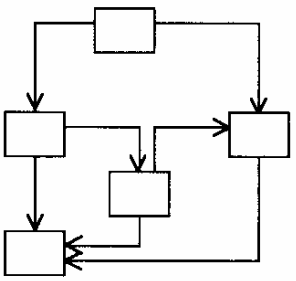
Information Structures

- Typical book-like structures, also useful for guided tours and systems like Guide:

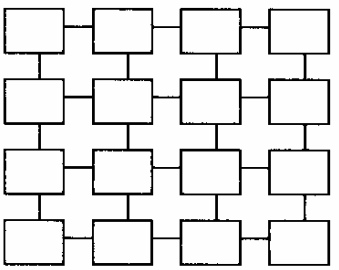


Information structures

- Networks are the typical hypermedia structure;
- A matrix is useful for highly structured information.



Network Structure



Matrix Structure

Functional requirements

- *navigability*: linking between nodes;
- *overviews*: structure maps, context;
- *information trails*: remember where the user has been, through history and bookmarks;
- *searching and indexing*: to find information;
- *document management*: authoring support;
- *presentation*: not necessarily on a screen;
- *customizability*: adaptation of content and links to individual users;
- *effective use of resources*: cpu, memory, network bandwidth;
- *handling of temporal media*: synchronization, interaction.

Nonfunctional requirements

- *link validity*: correctness, relevance, completeness, integrity; (examples follow)
- *content validity*: correctness, relevance, completeness, integrity;
- *concept organization*: organize the content into higher level concepts;
- *consistency and seamlessness*: consistent look and feel and seamless integration with other applications.

Non-functional link properties

- When links work and lead to a valid destination, they may still not be usable:
  - De link anchor must be meaningful. Many links have an anchor that reads **here**, or **homepage**, or some other meaningless word.
  - Even when the link anchor indicates the real destination, that destination must make sense to the user. Who knows where the following destination is?

Non-functional link properties

- **Structural links** provide the necessary navigation.
- **Cross-Reference links** (associative and referential links) provide "referral" to related information.
  - Too few xref links restrict navigational freedom.
  - Too many xref links cause confusion; they make selecting a link difficult.
  - Tests provide some evidence that a 50% ratio (structural vs. xref links) is optimal.

Non-functional requirements

- *Maintainability and evolvability*: applications need to be updated after an initial release;
- *reusability*: data and information often needs to be reused in other applications;
- *reliability and robustness*: the application must not crash, and data must always be available;
- *testability, validation, verification*: especially dynamically generated parts must be tested;
- *interoperability, portability*: the same information can be used with different hypermedia systems;
- *political and social aspects*: e.g. legal constraints;
- *cost-effectiveness*: keep development cost limited.

Tools

- *authoring tools*: authoring of nodes, links, and larger and high-level structures;
- *data manipulation tools*: preparing data for inclusion in an application (e.g. image manipulation, video editing, etc.);
- *analysis and design tools*: few tools exist to develop hypermedia applications in a systematic way;
- *process management tools*: no such tools exist for hypermedia development, but conventional tools may be adequate.

Analysis activities

- Client requirement analysis
- Content analysis
- User analysis
- Boundary analysis
- Implication analysis
- Constraint analysis
- Feasibility analysis

Client requirement analysis

- What is the purpose of the application?
- How will it be used?
- Who are the users?
- What content is already available, and in what form?
- What content must be developed, by whom?
- What are budget and time frame?
- What hardware/software is to be used?
- What level of security is required?

Content and boundary analysis

- Identify what content is required, what is available, and who will provide it.
- It is also important to decide the extent of the information: what information to include and what to omit.
  - In many cases the information scope is limited by what is available.
  - Some information may be restricted. Depending on the user some information cannot be delivered (and must not be hinted at).

User analysis

- Different aspects of the user (population) influence the “desired” hypermedia application:
  - computer literacy level / hypermedia experience
  - language literacy level (are the users native speakers of the application’s language)
  - cultural background (e.g. meaning of colors such as green and red)
  - expectations of what the application will do
  - age range, possible disabilities

Implication analysis

- Introducing the hypermedia application may have implications on the environment in which it is to be used.
  - The system may cause the loss of jobs.
  - The system may imply that employees need to provide information on-line in a shorter time frame than they are used to.
  - The system may introduce new ways of communication with customers.
  - etc.

Constraint analysis

- Different internal and external factors put limits on the application:
  - company policy (to keep certain information confidential)
  - standards: using standard technology (like HTML) may not be able to support certain desired functionality
  - the delivery platform (clients and servers) have limitations (regarding screen real-estate, processing power, network capacity, etc.)

Feasibility analysis

- Can the application be built with the given budget?
- Can the application be built within the given time frame?
- Can the desired functionality be achieved on the hardware/software platform of the users?
- When certain functionality cannot be achieved the requirements must be reconsidered.

Design activities

- Platform and environmental design
- Content design
- Information database design
- Access and navigation mechanisms
- Overall look and feel
- Guidelines for content development
- Installation engine design
- Authentication and logging

Platform and environmental design

- Very different platforms exist:
  - Kiosk application without keyboard or mouse (but with touch-sensitive screen).
  - Palmtop or WAP phone with very small screen.
  - Tightly integrated system on a workstation.
  - Web-based systems (to be viewed on a wide range of computing devices).
  - It is important to know the platform of the intended audience.
  - It is important to know the users' skills in installing software.

Content, structure and links

- Different design methodologies exist (HDM, OOHDM, RMM, ...)
  - There may be structure in the legacy data (from databases) already.
  - Database structures can be used to create structural links.
  - Designing associative links is more difficult:
    - from objects or concepts to applications
    - between cause and effect
    - between actors and actions
    - between whole and part (i.e. decomposition)

Access and navigation support

- Access through links alone is insufficient; a search and indexing facility is needed:
  - Full-text index: all words in the information sources are in the index;
  - Selected words: a statistical method can be used to select words, or structural elements can be used;
  - Keyword searching: allows arbitrary search terms, and applies statistical methods;
  - Control vocabulary: allow only certain words that have a well-defined meaning in the application domain.

Overall look and feel

- Screen layout:
  - Select appropriate metaphor, e.g. book metaphor;
  - Use familiar icons such as filing cabinet and trashcan;
  - Use appropriate labels for nodes and link anchors;
  - Use a “split screen”, e.g. through HTML frames, to dedicate parts of the screen to separate functions;
  - Design components such as forms, menus, buttons, icons, tool bars, dialog boxes, etc.
  - Select colors and fonts.

Content development

- Node construction (Horn, 1989):
  - **Chunking**: group information into manageable units (4 to 6 items in each unit);
  - **Relevance**: each node should be based on a single primary point or concept;
  - **Consistency**: for similar subject matters, use similar words, labels, etc.;
  - **Labeling**: proper labeling of nodes and groups of nodes helps users understand their context.

Information organization

- Identifying “good” nodes in existing information:
  - can often be done using structure of the information;
  - can sometimes be done using information retrieval techniques;
  - existing techniques do not translate easily to non-textual media.

Information integration

- In consistently structured information links may be generated automatically;
- Manually authoring links requires tools:
  - Many authoring tools have page and link editors that help create the **micro** structure;
  - Creation of the **macro** structure involves the integration of nodes, windows, forms, etc., into a cohesive application. Some tools to get an overview of the link structure exist (e.g. Macromedia Authorware and Microsoft Frontpage).

Conclusions on hypermedia design

- There is still no comprehensive approach which guarantees a successful design;
- The huge variation in problem, solution and development domains makes it unlikely that a single successful approach can be developed.
- Future work concentrates on the use of design patterns to identify successful application area dependent approaches.