

補講について

11/21 休講分の補講

情報学科CSコース
情報システム (3年後期)
第5回
(田島担当分第1回)

田島 敬史

2012 年 11 月 7 日

情報システムの歴史

Memex から Ajax まで

ここでいう情報システムとは...

- 人間の知的作業において情報を整理活用することを支援するためのシステム
- 社会における情報流通基盤となるシステム

最古の情報システム? (前者のタイプ)

- Memex (1945) — MEMory EXtender

今日, 最大かつ最も重要な情報システム (後者? 両方?)

- the World-Wide Web (1991)

ハイパーテキスト, ハイパーメディア, 構造化文書, Web

What is Web?

Web = HTTP + HTML + URI

Web = 分散ハイパーテキスト (分散ハイパーメディア)

HTML = ハイパーテキスト用構造化文書

What is 分散ハイパーテキスト?
What is 構造化文書?

ハイパーテキスト

What is ハイパーテキスト?

- ハイパーテキスト:
 - ノードとリンクからなるグラフ構造を持つデータ
 - 通常, ノードは文書(マルチメディア文書)
 - ノード中にアンカーを置きリンクで結んでデータ間の関連を表現(ノード全体をアンカーとする場合もある)
 - 非線型な閲覧を実現(\leftrightarrow 従来の線形文書の閲覧)
- 巻き物(scroll) \rightarrow 本(+目次, 索引) \rightarrow ハイパーテキスト

4

ハイパーテキスト

Dexter ハイパーテキスト参照モデル[†]

以下の三つの層からなるハイパーテキストシステムのモデル

1. 要素内部層: ノード内の部品となるデータの集合を蓄積
2. 蓄積層: アンカーを生成しグラフ構造を管理
同じ情報に対してもユーザに応じて異なるアンカーを生成
3. 実行時層: ユーザへの表示を決定

Web は, 1.5層?の構造からスタートして3層へ

Dexter モデルは, やっぱりよくできていた?

([†] Dexter は人名ではなく会合があったホテルの名前)

6

ハイパーテキスト

What is マルチメディア?

- マルチメディア:
 - テキスト, 画像, 動画, 音声などの複数種類のメディアからなる複合データ
 - 多くの場合, 二(or 三)次元空間上へデータを配置する構造と, 動画, 音声等の時間軸を持つメディアを含み, 空間上の配置と時間軸上の配置(特に同期)でデータの関連を表現
- ハイパーメディア:
 - マルチメディア + リンク

5

ハイパーテキスト

なぜ「情報システム」で歴史の勉強?

Robert Calliau[†] の講演 (at ACM Hypertext'99)
(不正確な記憶)

- 「Web は技術的には新しくなかった」
- 「当時のこの会議に論文を出したら落とされた」
- 「論文は落ちたが, デモをやったら, とても好評だった」
- 「この分野の過去の研究成果は宝の山であり, そこから宝を掘り出して, よいタイミングで世に出すことも重要」

([†] Tim Berners-Lee が CERN で初期の WWW を設計した当時の上司)

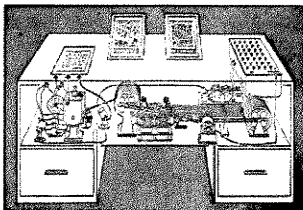
7

ハイパーテキスト

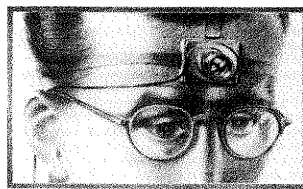
歴史 (1/2)

1945 Memex (Vannevar Bush: "As We May Think")

- あらゆる情報をマイクロフィルムで格納しリンクを設定可能
- 関連する情報を即座に取り出せる機械式高機能事務机
- 関連する情報にすぐアクセスできることで人間の思考活動を支援
- ハイパーテキストの源流



Memex is the heart of a desk world in which living files and material on any subject are subject to the operator's fingertips. Viewing translucent viewing screens magnify superimposed film by code symbols. At left is a technician which automatically printing alpha-numeric codes, pictures and letters, their files then in the desk for future reference (LIFE, 1945, p. 122).



A scientist of the future records experiments with a tiny camera fitted with universal-focus lens. The small square in the eyeglass at the left sights the object (LIFE, 1945, p. 122).

8

その他の関連する流れ

広域情報システム

- 80年代にインターネットが広まる
- 90年前後 Wais, Gopher 等のいくつかのシステム
- ftp + Archie
 - Archie = メジャーな ftp サイトにある全ファイルの索引
 - ftp + Archie = 登録型検索エンジン + Web
 - Web との大きな違いの一つは「リンクの有無」
 - リンクがないため明示的なサイトの登録が必要
 - ユーザ側にとっても、「ついつい色々なリンクをたどってしまっ
て...」ということは起こりにくい

10

ハイパーテキスト

歴史 (2/2)

1965 Project Xanadu (Ted Nelson)

- 「ハイパーテキスト」という言葉を提唱
- 世界規模のハイパーテキストの構想
- Web の源流

1987 HyperCard

- Apple Macintosh 上で動作した商用ソフトウェア
- Web 以前では、もっとも広く使用されたハイパーテキストシステム

1991 World-Wide-Web

9

その他の関連する流れ

Social Network Systems (SNS)

- Facebook, Twitter
 - タイムライン: 指定した全ユーザの投稿を時間順に表示。
 - 情報発信にもコミュニケーションにも用いられる。

関連する既存システム

- blog : 情報発信用. つながりが広がる仕組みに欠ける。
- chat : 非公開用. つながりが広がる仕組みに欠ける。
- RSS
 - 複数の情報源からの投稿を時間順に表示することも可能。
 - 他のユーザの subscribe 先を見ることは通常できず, SNS には
ならない。

11

その他の関連する流れ

オブジェクト指向(Smalltalk), ウィンドウシステム(Dynabook)

1973 Alto (Alan Kay, Dynabookのプロトタイプ)

- オブジェクト指向の概念に基づく Smalltalk を OS とする
- マウスによる「クリック」を用いた GUI インターフェイス
- オブジェクト指向の概念は多様なメディアを統一的に扱うハイパーメディアにおいては非常に重要な概念



12

HyperCard

Web との比較

- ノードに相当する単位は「カード」(または後述のスタック)
= Web の「ページ」
- カード内に、テキストフィールド、画像、動画、音声、ボタン等の部品を配置
- カード内のこれらの部品から他のカードへ「リンク」を設定可能
- 静的リンク: リンク先があらかじめ決定されていて一定
- 動的リンク: アクセス時にリンク先が計算される
↔ Web の CGI: サーバ側で動的にリンク先ページを生成
= JavaScript によるページ移動

14

HyperCard

HyperCard と Web の平行進化

HyperCard と Web を比較してみることによって、現在の Web を構成する様々な概念、技術を見てみる。

13

HyperCard

Web との比較

- スクリプト言語 HyperTalk
 - 部品にメッセージが届いた際に起動されるスクリプトを設定
= JavaScript
 - メッセージは「クリック」、「カードのオープン」等
 - 部品の移動、表示、非表示などができる
 - 「リンク」も実はスクリプトの一形式
↔ Web のハイパーリンク
- データ管理システムとしての用途とインタラクティブシステムとしての用途
= Web 上のアプリケーション、計算機管理システム、ゲーム等
- スタック = カードの集合(とそれらの間のリンク)

15

HyperCard

Web との比較

- フォアグラウンドカードとバックグラウンドカード
 - － 複数のフォアグラウンドカードで一つのバックグラウンドカードを共有
 - － 「一つの書式と、それを用いる複数のデータ」の表現などに使える
 - ≡ Web のスタイルシート

16

構造化文書

二種類の文書の構造

- 論理構造: 文書の論理上の構成を表すもの
例: タイトル, 章, 段落
- 物理レイアウト構造: ある特定デバイス上での表示上の構造
例: ページ, フォント, 段落間のスペーシング

論理構造だけ記述してレイアウトを自動生成したい

18

構造化文書

SGML と HTML

構造化文書:

文書の構造の情報を機械が理解可能な形で埋め込んだもの

SGML (ISO 標準)

- 様々な「マークアップ言語」を定義するための枠組み(メタ言語)
- 大企業での文書管理(e.g. 仕様書, 飛行機の整備マニュアル)
- タグの省略などの高度な機能

HTML

- ハイパーテキストマークアップ言語
- SGML のインスタンス
- `</p>` の省略などが可能

17

構造化文書

二種類の文書の構造

- 例1:
`<p>`タグは「段落」という論理構造を表し, それを「改行」で表すのか, 「改行+空行」で表すのかというレイアウトは自動生成される.
- 例2:
`<h1>`タグは「大見出し」という論理構造を表し, それを「太字」にするのか, フォントサイズや配置はどうするのかというレイアウトは自動生成される.

19

構造化文書

レイアウトの生成方式

手動生成:

- ユーザがレイアウトを完全に指定
- DTP(desktop publishing)ソフトなど高度なレイアウトを必要とする物
- WYSIWYGアプローチ(What You See Is What You Get)

自動生成:

- ユーザは完全に論理構造のみを指定
- 例: $\text{T}_{\text{E}}\text{X}$ (Donald Knuth), $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ (Lesile Lamport)
各文書ファイル + (ほぼ固定の)スタイルファイル
- 統一された書式を持つ多量のデータに有効

20

構造化文書

スタイルシート

HTML では,

- HTML 規格で定められたタグを使い, 論理/表示構造を記述
- あるレベルでのクライアントデバイス独立を実現
(e.g. テキスト端末での表示, ウィンドウの大きさ)
- WYSIWYG と論理記述の中間?
- 自動生成 \leftrightarrow もう少し指定したければスタイルシートを記述
CSS(Cascading Style Sheets)

論理構造とレイアウト情報をさらに分離する方向

→ 後述のXML + XSL

22

構造化文書

レイアウトの生成方式

半自動生成:

- ユーザがデータの中に以下の二つを分離して記述
 1. 論理構造を記述したデータ
 2. レイアウト規則
- 同じレイアウトの繰返しが現われる文書に有効

21

構造化文書

HTML から XML へ

HTML

- 規定されたタグのみ. マクロなどの機能もなかった.
- 人間が読む文書の記述を想定
- 相互運用性が低い
(SGMLはタグの省略機能など高度すぎて処理系の実装が大変)

XML

- 独自のタグを定義可
- 人間による閲覧だけでなくプログラムによる利用へ(e.g. Google, Amazon)
- 簡素な仕様

23

構造化文書

歴史 (1/2)

1986 SGML

1991 HTML 1.0

- タグさえないシンプルなもの

1995 HTML 2.0

- 一般ユーザへの普及、レイアウト機能の充実。
, <form>

1997 HTML 3.2

- イン트라ネット等でのインタラクティブシステムとしての利用
- <table>, <script>, <applet>

24

より高度なリンク機能: XLink

概観

HTML の <A> タグによるリンク

- 二つのリソース間の片方向のリンクを記述
 - 始点となるリソースにリンクを記述
 - 終点となるリソースを URI で指定
- リンクの効果、動作などは記述できない

26

構造化文書

歴史 (2/2)

1997 HTML 4.0

- スタイルシート

1998 XML

2000 XHTML 1.0

- HTML の XML 版 (従来の HTML は SGML 版)

2014? HTML5, XHTML5

- インタラクティブなアプリケーションのための機能の充実

25

より高度なリンク機能: XLink

概観

XLink (<http://www.w3.org/TR/xlink/>)

- 様々なデータ間のリンク付けを記述するための規格
- 当初, XML データのためのリンク機構として XML 規格の一部となる予定だったものから発展
- リソースの指定には, やはり URI を使用
- 任意個のリソース間の, 任意の方向のリンクを定義可能
- リンク定義をリソース自身とわけて外部に記述可能
- リンクの効果 (タイミング, 動作内容) を記述可能

27

XLink用語

リンク:

- 複数のリソース (またはその部分) 間の関係を記述するもの

リソース:

- XML データに限らない
- リモートリソース = URI によって指定されるもの
- ローカルリソース = その要素自身の属性で指定

トラバーサル:

- リンクの利用 = トラバーサル
- 必ず始点リソース一つと終点リソース一つを含む

28

XLink構文

- 任意の要素に対して属性を付加することでリンク要素を定義
- 既に存在する属性名と重ならないように namespace を使用
- 以下の物からなる
 - リンク要素: simple と extended の二種類
 - locator: リモートリソースを URI で指定
 - resource: リンク要素の子要素をローカルリソースに指定
 - アーク定義: 始点, 終点, 動作, 動作タイミングを指定
 - * 動作: new(別表示), replace(表示切替), embed(埋込), 等
 - * タイミング: onLoad(始点を読み込まれたら即), onRequest(ユーザの要求(クリック等)に応じ)

30

XLink用語 (続き)

アーク:

- トラバーサルの方向, トラバーサルの際の動作等を記述したもの
- 一つのリンク中に複数のアークを定義可能
- outbound: 始点がリモート, 終点がローカル
- inbound: 始点がリモート, 終点がローカル
- third-party: 始点がリモート, 終点もリモート

29

XLink複雑なリンク記述の例

```
<myLink xmlns:xl="http://www.w3.org/1999/xlink" xl:type="extended">
  <person xl:type="locator" xl:href="http://a.org/~kay" xl:label="kay" />
  <photo xl:type="locator" xl:href="http://b.com/ox.jpg" xl:label="ox" />
  <review xl:type="resource" xl:label="review">
    <score>4.0</score>
    <comment>best among his early works</comment>
  </review>
  <by xl:type="arc" xl:title="photo by" xl:from="ox" xl:to="kay"
    xl:show="replace" xl:actuate="onRequest" />
  <rev xl:type="arc" xl:title="my review" xl:from="ox" xl:to="review"
    xl:show="new" xl:actuate="onLoad" />
</myLink>
```

31

SMIL (<http://www.w3.org/TR/smil20/>)

- IE, Real Player, Windows Media Player 等がサポート
- マルチメディアデータのプレゼンテーションのシナリオを記述
 - レイアウト: 画面上の配置
 - タイムライン: 各要素データの表示の開始, 終了, 持続時間
 - * `begin="sIcon.click+2"`
 - 「sIcon」要素がクリックされた2秒後に開始
 - * `end="1:30"` → ページのロードから1分30秒後に終了
 - 同期関係
 - * `<par>` → 並列再生
 - * `<seq>` → 逐次再生
 - ハイパーリンク

SMIL

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head><layout>
    <region id="title" height="%50" />
    <region id="main" top="%50" />
  </layout></head>
  <body>
    <par>
      <seq><video src="start.rm" region="title" />
        </seq>
      <a href="http://a.com/another.smi" show="new">
        </a>
      </par>
    </body>
  </smil>
```

