

木構造

総合科目「ネットワーク社会を支える情報技術入門 II」、2009年09月24日

筑波大学システム情報工学研究科 / 電子・情報工学系
新城 靖
<yas@cs.tsukuba.ac.jp>

このページは、次の URL にあります。

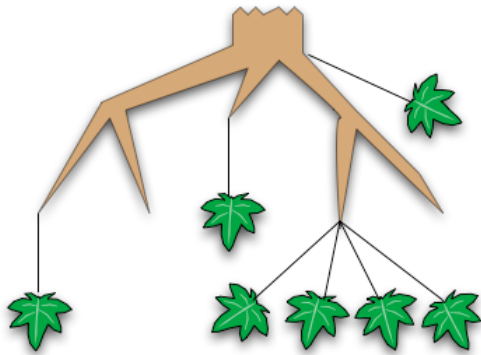
<http://www.softlab.cs.tsukuba.ac.jp/~yas/gen/it-2009-09-24>

あるいは、次のページから手繰っていくこともできます。

<http://www.softlab.cs.tsukuba.ac.jp/~yas/gen/>

<http://www.cs.tsukuba.ac.jp/~yas/>

印刷配布資料 <http://www.softlab.cs.tsukuba.ac.jp/~yas/gen/it-2009-09-24/it-2009-09-24.pdf>



■今日の重要な話

- 情報を整理する時に、もっとも基本的で重要な方法として、木構造を使う方法がある。
- 木構造には、様々な表現方法がある。
 - 節と枝。根も葉もある。根を上を書く。
 - 領域的(集合のベン図)
 - 段付け (字下げ)
 - (区切り文字を挟んで)単語を並べる
- 木構造は、コンピュータであつかうものの名前を管理するために使われている。
- 木構造は、英語の読解や作文に役立てることができる。
- 木構造を使うと要素の数が増えても対応することができる。
- 木構造だけでは、情報の整理はうまくいかない。木構造を補う仕組みが必要になる。
 - ファイル名では、非循環グラフ (エイリアス、ショートカット、シンボリック・リンク) が使われている。

- World Wide Web では、ハイパーテキスト(グラフ構造)が使われている。
- URL (Uniform ResourceLocator) には2つの木の表現が含まれている。

■木構造

木構造 (tree structure) というのは、コンピュータ・サイエンス (情報学類、情報科学類で学ぶ学問) でよく使われる用語である。分野によっては、同じものを**階層構造 (hierarchical structure)** という言葉で表現することが好まれる。ドイツ語語源の、ヒエラルヒー(Hierarchie)という言葉が使われることもある。

木構造の例を、大学の組織を使って説明する(図1)。

- 木構造は、節と枝からできている。
- 特別な節として、**根(root)**が1個ある。
- 末端の節を**葉(leaf)**と言う。
- 木構造では、ある節から別の節までの道が1通りしかない。
- つながっている節と節で、根に近いものを**親**、そうでないものを**子**という。

木構造という名前は、本物の木が、一度枝分かれした後は決して交わらないことに似ていることによる。ある節から別の節までの道が2通り以上あるは、**グラフ構造**と呼ばれる。

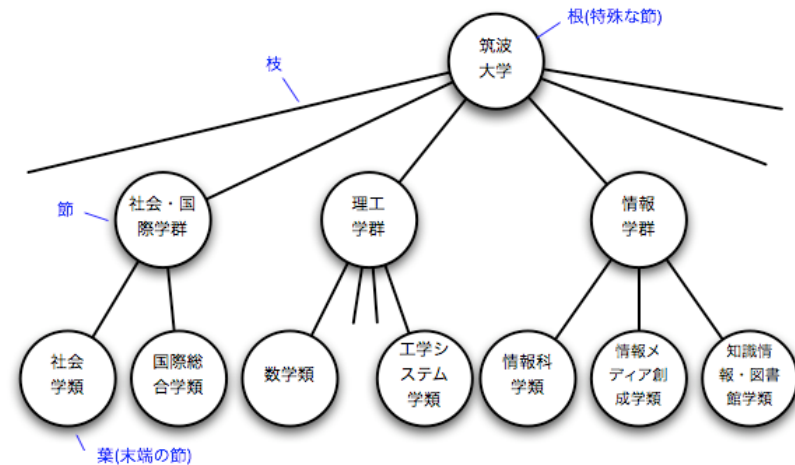


図1 大学組織に見られる木構造

- 筑波大学が木の根にあたる。根からは、何本かの学群の枝が出ている。コンピュータ・サイエンスでは、木の根を上を書く習慣がある。
- 情報学群の節には、情報科学類、情報メディア創成学類、知識情報・図書館学類の子の節がある。情報学群の親は、筑波大学である。
- 学群は中間節、学類は葉(端節)である。

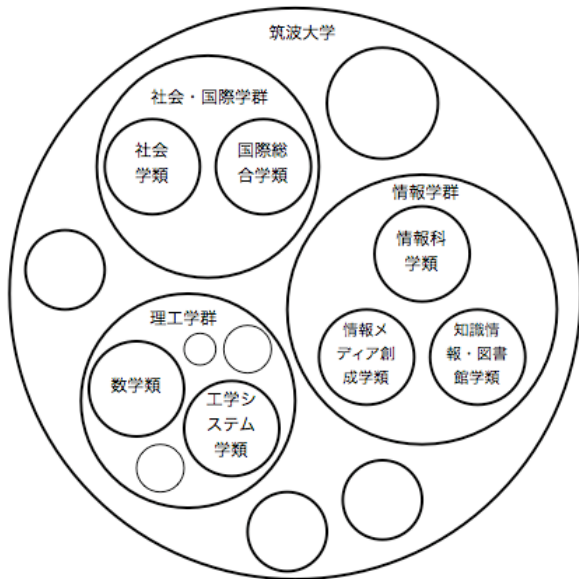


図2 大学組織に見られる木構造（領域的な見方）

A、B、2つの集合があると、一般には、次の4つに分割される。

1. Aにしか含まれない部分
2. AとBの両方に含まれる部分
3. Bにしか含まれない部分
4. AにもBにも含まれる部分

領域（木構造）の場合、完全に含まれるか、無関係かのどちらかになる。

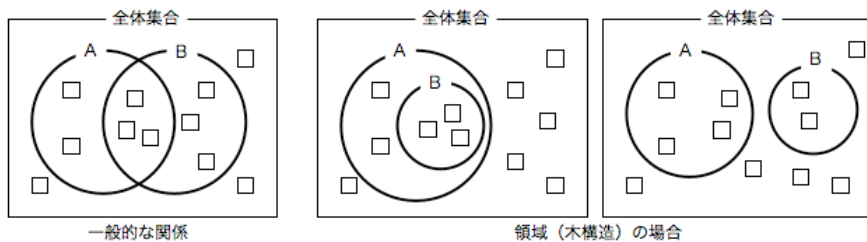


図3 2つの集合の関係

◆字下げによる木の表現

木構造を字下げで表すことがある。

筑波大学

-
- 社会・国際学群
 - 社会学類
 - 国際総合学類
- 理工学群
 - 数学類
 -
 - 工学システム学類
- 情報学群
 - 情報科学類
 - 情報メディア創成学類
 - 知識情報・図書館学類
-
-

◆区切り文字入り表記

「情報科学類」という節を、次のように表記する。

筑波大学情報学群情報科学類

コンピュータの中で、文字列（文字の並び）で木構造上の位置を表現する時には、節が分かりやすいため、はっきりと区切りを入れて表現することがよく行われる。

筑波大学 情報学群 情報科学類

筑波大学／情報学群／情報科学類

情報科学類 情報学群 筑波大学

区切り文字としては、「.」（点）、「/」（スラッシュ）、「\」（バック スラッシュ）、「¥」（円記号）などがよく使われる。単語を並べる時に、木の根に近いほうから書く流儀と遠い方から書く流儀がある。

◆木の例

コンピュータでは、次のような場所で木構造が使われている。

- ファイルの名前
- インターネット上のコンピュータの名前
- 電子メールのアドレス
- World Wide Web のアドレス (URL)
- ネットワーク・ニュースのニュース・グループの名前

- さまざまなメニュー

コンピュータ以外では、次のような場所で木構造が使われている。

- 分類（生物、図書館の本）
- 住所
- 会社組織、官僚機構
- 1つの文
- 1つの文書

◆文の構造

1つの文も、木構造で表される。同じ単語の並びでも、組み立てられる木構造が違くと違う意味になる。英文の解釈は、木構造を組み立てることである。

Time flies like an arrow.

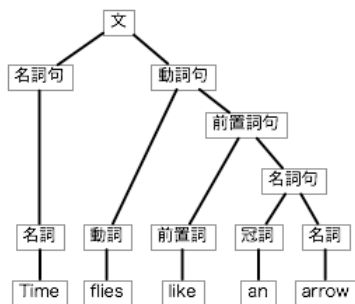


図4 「Time flies like an arrow.」の木(その1)

光陰矢のごとし。

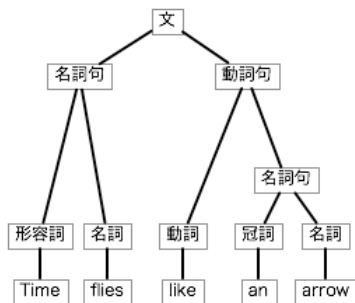


図5 「Time flies like an arrow.」の木(その2)

時蠅、矢を好む。

◆英語の文法

基本5文型は、木構造を意味する。

1. S V
2. S V C
3. S V O
4. S V O O
5. S V O C

これを <clause> とする。英語の文は、次の形式で <clause> をいくつかつなげ、最後に「。」をつけたものである。

```
<sentence> ::= <simple sentence> '.'
              または <compound sentence> '.'
              または <compound-complex sentence> '.'
```

```
<simple sentence> ::= <clause>
```

```
<compound sentence> ::=
  <clause> ',' <FANBOYS> <clause>
  または <clause> ';' <transition word> <clause>
  または <clause> ',' <conjunctive> <clause>
```

```
<complex sentence> ::=
  <dependent clause> ',' <independent clause>
  または <independent clause> <dependent clause>
```

```
<compound-complex sentence> ::=
  <clause> <dependent clause>
```

```
<dependent clause> ::= <subordinate> <clause>
```

```
<subordinate> ::=
  <subordinate-adj> または <subordinate-adv>
  または <subordinate-noun>
```

```
<subordinate-adj> ::=
  who または whom または that または which
```

```
<subordinate-adv> ::= after または before または because
  または although または since
```

```
<subordinate-adv> ::= that または whether
```

```
<clause> ::= 基本5文型で表せるもの
```

```
<FANBOYS> ::=
  for または and または not または but または
  or または yet または so
```

```
<conjunctive> ::=
  <subordinate conjunction> または <coordinate conjunction>
```

または <conjunctive adverb>

<subordinate conjunction> ::= as, if, that など

<coordinate conjunction> ::= and, but, or, for など

<conjunctive adverb> ::= however, nevertheless,
still, then など

よい英文は、1つの <sentence> には、2つの <clause>、つまり、S V が2つが現れることが多い。

◆文書の木

文章の中にも、木構造が現れる。

- 章立て。章、節、項。
- 箇条書き。箇条書きの中に箇条書き。
- パラグラフ

◆パラグラフの木

英語国民に対する英語の教育では、パラグラフ(段落)の内部も、きちんと木構造で書くことが求められる。

- パラグラフは、1つの topic を語る (0個、2つ以上は不可)。
- 1つのパラグラフには、1つの topic sentence (パラグラフの内容を一言で語った文)を置く。
- topic sentence は、多くの場合、パラグラフの先頭に置かれる。
- パラグラフに含まれる他の文は、次のいずれか (宙に浮いているものは不可)。
 - topic sentence を展開させたもの (topic sentence を木の根として、その子供の節になる)
 - (他のパラグラフとの関係を示すもの)
- 木構造を明確にするために、signal wordsを使う。
 - First, Second, Third, ...
 - 原因・結果
 - 対句、対比

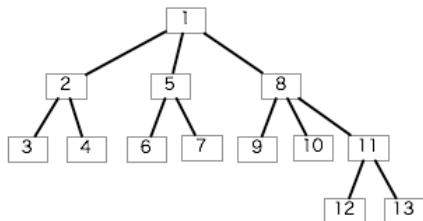


図6 topic sentence を根とする木構造

1,2,3 は、文の順番。木の根が上。

(よく書かれた) 英語の長い文章を斜めに読むには、パラグラフの先頭の topic sentence だけを読めばよい。

◆日本語の段落の構造:逆茂木型

逆茂木: よろい、かぶとの時代に、敵の侵入を防ぐために樹を斬り倒して尖った枝を外に向けて並べた障害物。

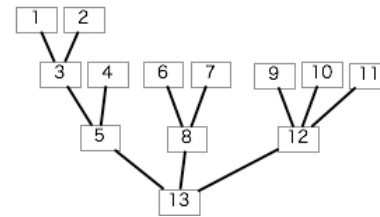


図7 逆茂木型の段落

1,2,3 は、文の順番。木の根が下。

日本語の特徴

- 漢字仮名混じり文。重要な言葉が漢字で書かれている。
- 文の構造。動詞が最後。

(日本語ではなくて) 「国語」の教師は、木構造 (生成文法、文脈自由文法、キヨムスキー(学者の名前)) を知らないこともある。

日本語でも、文学作品をのぞいて、英語的に木構造になっていると読みやすい。

◆is-a関係

This is-a pen の意味。

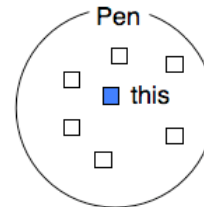


図7 「This is-a pen」の集合的な意味

is-a関係

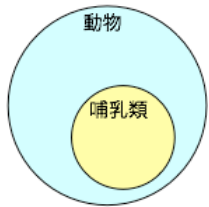


図8 is-a関係の例(哺乳類 is-a 動物)

◆has-a関係

工業製品などが部品で構成されている。

例: パソコン。

- ソフトウェア
 - システム・ソフトウェア
 - オペレーティング・システム
 - コンパイラ
 - アプリケーション・ソフトウェア
 - ワードプロセッサ
 - Web ブラウザ
 - メール・リーダー
- ハードウェア
 - 本体
 - マザーボード
 - CPU
 - メモリ
 - ハードディスク
 - 電源
 - キーボード
 - ディスプレイ
 - マウス

◆ディレクトリの木

ディレクトリは、全体では 木構造(tree structure) になっている。階層化ディレクトリ (hierarchical directory) と呼ばれることもある。

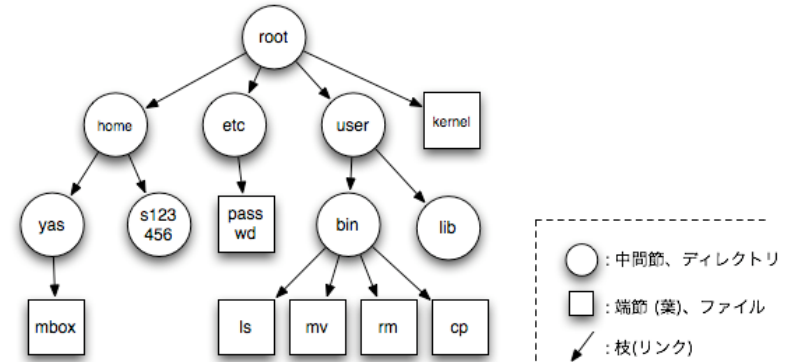


図9 ファイルとディレクトリの木

ファイルは、葉(leaf)になる。ディレクトリは、節(node)になる。特殊な節として、根(root)がある。これを、ルート・ディレクトリ (root directory) という。

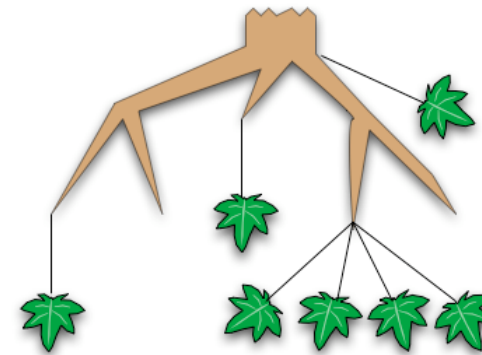


図10 自然の木

◆パス名

ファイルの名前の表現には、「パス名」がよくつかわれる。パス (path) というのは、道の意味である。パス名では、どの道を通ればよいかの道順を示すことでファイルの名前を表現する。

木構造では、節、または、枝(道)に名前がついている。ファイル名は、区切り文字で区切られた、節、または、枝の名前の並びになる。ファイルの名前を表現する時の区切り文字としては、次のものがよく使われる。

- /
- \
- :

ファイル名で「.」は、木構造の区切りとしては使われない。

パス名の例:

- /home/s123456
- /usr/bin/emacs

◆絶対パス名と相対パス名

パス名には、次の2種類がある。

絶対パス名 (absolute path name)

絶対パス名というのは、ディレクトリの木の根(ルート・ディレクトリ) から出発する道順

相対パス名 (relative path name)

現在着目しているディレクトリ (カレント・ワーキング・ディレクトリ) から出発する道順

例: 絶対パス名 /usr/bin/awk

1. ルート・ディレクトリから出発する
2. usrという枝へ進む
3. binという枝に進む
4. 最後に (awk)という枝に進む。

よって、/usr/bin/awk は、こういう手順で見つかるファイルを意味する。

例: 相対パス名 bin/awk (カレント・ワーキング・ディレクトリが /usrの時)

1. カレント・ワーキング・ディレクトリから出発する
2. binという枝に進む
3. 最後に (awk)という枝に進む。

◆ホーム・ディレクトリ

複数の人が使うコンピュータで、個人のファイルを保存する時の出発点となるディレクトリを、ホーム・ディレクトリと呼ぶ。

たとえば、icho という名前のコンピュータで、新城のホーム・ディレクトリは、絶対パス名では、/home1/yshinjo/ である。

- ルートディレクトリ (/) から出発する。
- home1 という名前の節に進む。
- yshinjo という名前の節に進む。

◆世界最大の木構造:DNS(Domain Name System)

電子メールを送ったりWorld Wide Web のページを閲覧する時には、データの 送り先やデータを持っているコンピュータを指定する必要がある。インターネットで使われている、コンピュータの名前を管理する仕組みは、DNS(Domain Name System) と呼ばれている。DNS では、膨大な数のコンピュータの名前を含む名前空間を階層的にドメイン (領域) に分割して管理している。

たとえば、次のような名前を考える。

azalea1.coins.tsukuba.ac.jp

このように、インターネットでのコンピュータの名前は、「.」で区切られた文字列 (文字の並び) である。この文字列で使える文字は、アルファベットと数字、ハイフン (マイナス) である。DNS で名前は、右から左に向かって解釈される。

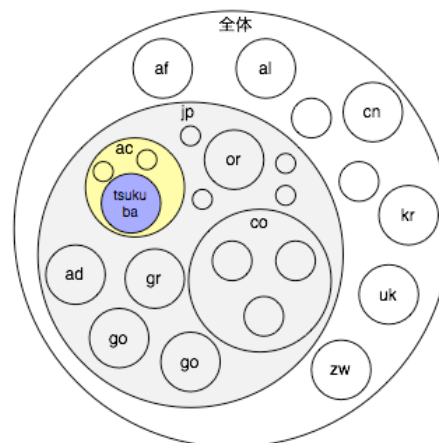


図8 名前空間のドメインへの分割

「azalea1.coins.tsukuba.ac.jp」を 図8 で考えると、次のようになる。

- 名前空間全体は、まず、af, al, . . . , jp, . . . , uk, zw というドメインに分割されている。その中の jp を選ぶ。
- その中の jp ドメインは、ac, ad, co, ne, gr, or などのドメインに分割されている。その中の ac を選ぶ。
- その中の ac ドメインは、ass, . . . , tsukuba, . . . , zokei 等のドメインに分割されている。その中の tsukuba を選ぶ。
- tsukuba ドメインは、さらに cc, is, coins などのドメインに分割されている。その中の coins を選ぶ。
- 最後に、coins ドメインに azalea1 という名前が登録されている。

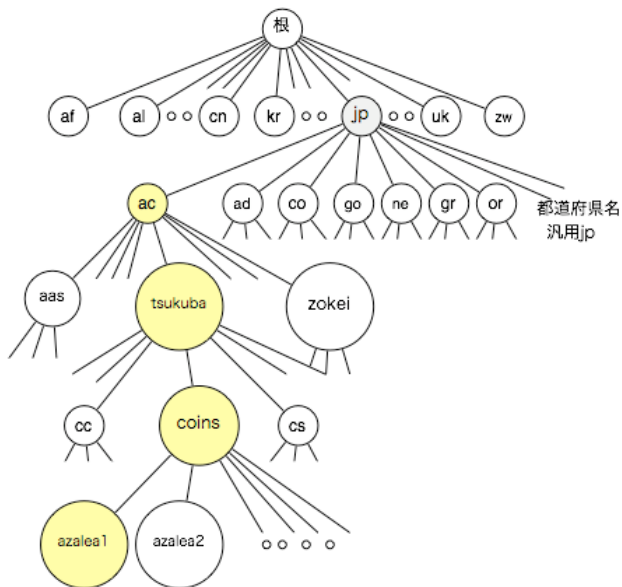


図9 名前空間の木構造としての見方

「azalea1.coins.tsukuba.ac.jp」を [図9](#)で考えると、次のようになる。

- まず根から出発する。
- 根の下の jp という節を選ぶ。
- その下の ac という節を選ぶ。
- その下の tsukuba という節を選ぶ。
- その下の coins という節を選ぶ。
- 最後に、その下の azalea1 という節(葉)を選ぶ。

◆DNSができるまで

初期のインターネットは、コンピュータの名前（ホスト名）は、フラットな名前空間が使われていた。

問題

- 名前の衝突。同じ名前を取り合いになった。
- 中央の管理組織の負荷が増えた。
- 最新のホスト名の一覧表の維持が困難になった。コピーがネットワークを重たくした。

1986年、3100の公式名と6500の別名。

1990年、6400の公式名。DNSに移行。この時点で、137,000。

2009年9月ごろの名
の統計

ドメイン名	数
.com	2133万
.de	546万
.uk	335万
.net	308万
.org	233万
.jp	110万
.info	82万
.it	68万
.biz	67万

.uk は、.co.uk, .org.uk, .sch.uk, .ltd.uk, .me.uk, .plc.uk の合計。 .jp は、属性型、地域型含む。

第二レベルのドメインの数。ホストの数はもっと多い。

<http://jpinfo.jp/stats/> JPドメイン名に関する統計 by JPRS <http://www.domainworldwide.com/>
The Daily Domain Counts of Domains Worldwide by Domain Worldwide.

◆ICANN

ICANN (Internet Corporation for Assigned Names and Numbers)

インターネットにおけるドメイン名や IPアドレスを調整する ための非営利法人。1998年に設立。

◆Top Level Domain(TLD)

木構造で考えて、根の直下に位置する節に相当するドメイン。ICANN が制約している。

◆TLDの種類

ccTLD (country code Top Level Domain)

ISO (国際標準化機構, International Standardization Organization) が定めた2文字による国別コード(country code) ([ISO 3166](#)) を使う。日本の国別コードは、jp。

gTLD (generic Top Level Domain)

ccTLD 以外。伝統的: .arpa, .com, .edu, .gov, .int, .mil, .net, .org。新しい: .aero, .biz, .coop, .info, .museum, .name, .pro (2000年)、.jobs, .travel, .cat カタロニア語, .mobi, .asia, (2005年)、.tel (2006年)

<http://www.icann.org/en/tlds/>.

◆TLDの自由化

2008年6月に、ICANN は、2009年から TLD を自由化する計画を発表した。英語の37文字(a-z,0-9,-)以外の文字(日本語も含む)でも利用可能にする予定。手数料 US\$18,750。商標との衝突など解決すべき問題も多い。(2009年9月現在、最終版ガイドブックを公開中。)

<http://www.icann.org/en/announcements/announcement-4-26jun08-en.htm>

<http://www.icann.org/ja/announcements/announcement-4-26jun08-ja.htm> (日本語)

■木構造の制約と問題点

大量の情報を保存するには、木構造を使うしかない。しかし、、、

◆こうもりの分類問題

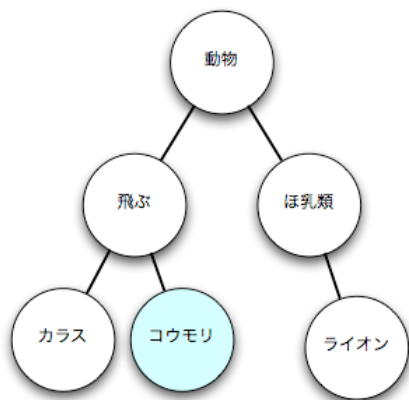


図13 こうもりの分類 (1)

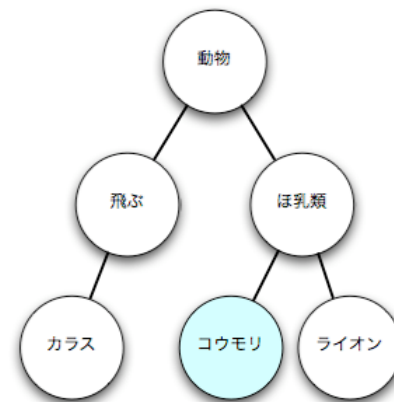


図14 こうもりの分類 (2)

◆シンボリック・リンク/エイリアス/ショートカット

木構造は、ファイルを整理するのに非常に強力な構造である。しかし、それだけでは、ファイルを整理するには不都合が起きる。それを解消するために、次のような名前と呼ばれる仕組みが用意されている。

- エイリアス(alias, Macintosh)
- ショートカット(shortcut, Windows)
- シンボリック・リンク(symbolic link,UNIX)

2つの節に、「別名」をつけて、2つの道からたどり着けるようにする。(木構造では、1つの節にたどり着く道は、ただ1つしかない。)

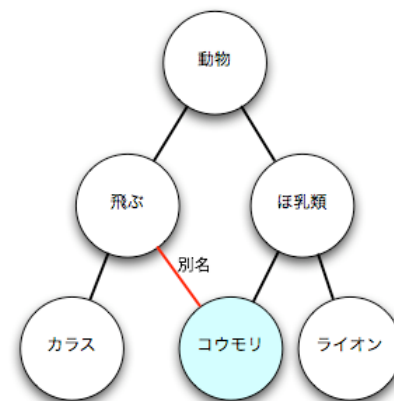


図15 こうもりの分類 (別名つき)

◆図書の分類

10進分類が多く使われる。

- 木構造の各節は、(最大)10個の子供の節への枝を作る。枝には、0から9の名前をつける。
- 根から枝をn個たどった場合、n個の数で表す。例: 007.64 は、根から 0→0→7→6→4 とたどった枝を意味する。

図書の分類の問題点

- 木のバランスがとれていない。時代の変化について行けない。兄弟の節で、自分より下の節の本の数が違う。(例: コンピュータ系は、数が多いが、007.6 のように深い階層にある。)
- いくつかの種類がある。日本十進分類法、デューイ十進分類法。
- 1つの視点からの分類しかできない。コウモリの分類問題問題。

◆木構造を使わない図書の分類

インドの図書館学者S. R. Ranganathan により考案された方法。5つのファセット(カテゴリ)を使う。

◆ファセット(カテゴリ)をつかう分類

木構造とは違い、順番を気にしない。

例: アパートの分類。次のような項目から任意の順で絞り込める。

- 住所
- 家賃
- 間取り
- 通勤時間
- 築年数

◆Folksonomy(フォークソノミー)

「folks (大衆) + taxonomy (分類法)」

情報(Webページ、ブログなど)に、各人がタグを付けていく。そのようなタグ付けされた情報を多数持ち寄ると、大衆(folks)に指示されたタグが浮かび上がる。

Social Bookmark などによく使われている。

◆二分木(binary tree)

子供の節の数が最大2個(0個でも1個でもよい)。(左右が区別される。)

◆系統樹(phylogenetic tree)

生物の進化を、最初の生物を根とする木構造で表したものの。

- 普通は、二分木で書く。(分岐した時期が近くなるとどちらが先かよくわからないものもあるが、必ずどちらかが先と考える。)
- 種Aと種Bの共通の祖先となるものが存在する。(今は絶滅したかもしれないが。)

参考: <http://www.biol.tsukuba.ac.jp/tjb/Vol5No9/TJB200609TM.html>

岡岡 俊樹(元 筑波大学 生物科学系): "動物系統分類学—私の昔のテキストから—", つくば生物ジャーナル Tsukuba Journal of Biology (2006) 5: TJB200609TM.

◆細胞内共生説

もともと別の種であったものが、合体して1つの種になる。例:

- ミトコンドリア
- 葉緑体

これを考えると木構造ではなくなる。

◆マルチ商法

Multi-Level Marketing。木構造の悪用の例。

- 各人が節になる。
- 他の人を勧誘して、子供の節にする。

各人がn人を勧誘したとすると、mレベルの人数が n^m 人となる。n == 2としても、m == 27で、参加者が日本の人口を超え、新たな参加者がいなくなる。子供の節の売上げが親の利益につながるとしても、潤うのは根に近い一部の節だけ。

◆官僚制度の2つの見方

- 整理棚。人間を系統的に分類する。事実を系統的に分類する。(保存)
- 情報が流れる経路。(通信)

情報の流れには予算の流れも関連している。

- 政府の予算のトップレベルは、国会の審議で省庁に分割して割り当てる。
- 一度省庁に分割した予算を、下流で合算して使うことはできない。(これを許せば、国会審議の意味がなくなる。)

◆官僚制度に見られる木構造の問題点

- 大きさ、細分化の問題。縦割り行政（木構造行政）。
- 時間的に古くなる。環境変化についていけない。日本の財務省/国会は、省庁予算に「シーリング」を設定して、省庁に配分を任せた。予算配分が固定化されてしまった。

中間管理職の意味＝横方向に情報が流れない。

木構造でしか情報が流れないような組織は、潰れる。木構造を補う意味で、会社組織では、裏チャンネルや同期会が重要となる。

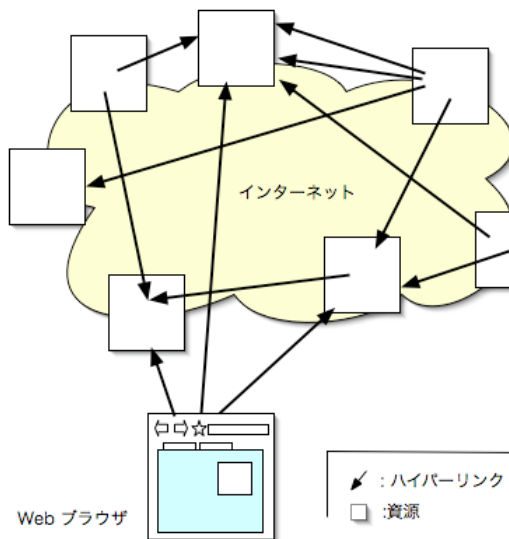
日本政府の「国家戦略局」が木構造を打破できるか。

■ハイパーテキストとハイパーメディア

木構造を補う方法として、ハイパーテキストと呼ばれる方法を使うことがある。

ハイパーテキスト(hypertext)とは、内部に他のテキストへの「参照(reference)」が埋め込まれているテキスト(文書、文字だけから構成されるデータ)である。ハイパーテキストという仕組みを使えば、テキストのある部分から、関連している情報を含んでいるテキストのある部分を引き出すことが簡単になる。

ハイパーテキストを拡張し、テキスト・データだけでなく、音声や画像などのデータを扱えるようにしたものを、ハイパーメディア(hypermedia)という。World Wide Webは、(木構造ではなく)ハイパーメディアに基づいて作られている情報提示のための仕組みである。



HTML や SGML では、「<」と「>」で括られた範囲がタグになる。タグには、「<name>」という形式と「</name>」という形式の2つの種類があり、前者を 開始タグ 後者を 終了タグ といいます。開始タグと終了タグに囲まれた部分が、マークが付けられたテキストになる。

ハイパーリンクを実現するためのマークアップ言語には、次の2つの機能が必要になる。

- データにラベルを付ける(参照の目標となるマークを付ける)。
- データにラベルへの参照を埋め込む。

◆URL

HTML では、他のデータへの参照を実現するためにURL (Uniform Resource Locator) という形式を使う。次に、URL の例を示す。

`http://www.tsukuba.ac.jp/education/college.html`

http

HyperText Transfer Protocol. WWWのデータを保持しているプログラムと、WWWを表示するプログラムの間でデータをやり取りするときの形式を定めた約束。

www.tsukuba.ac.jp

そのデータを持っているコンピュータの名前。

/education/college.html

そのコンピュータの中での資源の名前(ファイルの名前)。最後の.htmlは、その資源がHTMLで書かれている事を表わしている。

◆URL中の2つの木

URLには、2つの木構造の表記方法が混じっている。

- 木の根が左(ファイル名)
- 木の根が右(コンピュータの名前)

◆URLを間違えた時/URLが変更された時

2種類のエラー

- Not Found. The requested URL **XXX** was not found on this server.
- Unable to locate the server URL **XXX**. Please check the server name and try again.

長いURLのどの部分が怪しいかを区別できるようにする。

エラーが出た時には、木構造で親を探してみる。

Last updated: 2009/09/16 18:53:14

[<Yasushi Shinjo / <yas @ cs.tsukuba.ac.jp>](mailto:Yasushi.Shinjo)