

# タグ付け文書からの多様な概念図の生成

村山 正司      中村 裕一      大田 友一

筑波大学 機能工学系

〒 305-8573 茨城県つくば市天王台 1-1-1  
E-Mail: murayama@image.esys.tsukuba.ac.jp

本稿では、文書の概観性を高めるために、文書の内容を表現する概念図を自動生成する手法について紹介する。この手法ではタグ付け文書を入力として想定し、意味的性質を記述するタグを設定する。タグの意味的性質と概念図の構造とをうまく対応づけることにより概念図を生成する。ここではユーザの好みに応じて多様な概念図を生成するため、複数の対応のパターンを選択できるようにした。また一般的な文章にこのタグを付加し、概念図を生成する実験を行った。

## Flexible Diagram Generation From Tagged Text

Masashi Murayama      Yuichi Nakamura      Yuichi Ohta

Institute of Engineering Mechanics and Systems, University of Tsukuba  
1-1-1 Tennoudai, Tsukuba, 305-8573, Japan  
E-Mail: murayama@image.esys.tsukuba.ac.jp

We often need too much time for reading documents, since it is often difficult to efficiently grasp their outline. For this purpose, we propose our diagram generation scheme for presenting the structures of a text. The semantic structure of a tagged text is effectively translated to various diagrams, and they are linked to the text. In this paper, we first describe how semantic structures can be expressed by a diagram. Then, we propose our framework for automatic diagram generation from tagged texts to diagrams.

### 1 はじめに

近年、インターネットの普及に見られるように、膨大な情報へ簡単にアクセスできる環境が整ってきた。しかし一方でデータ量が大きい情報洪水と形容される問題を引き起こしている。

特に、文書は一次元メディアであるという性質から、内容を把握するのに時間がかかる。たとえば、文書中で離れた語句の間にある意味的な関係などは、文書を一読しただけでは分かりにくい。文書

の持つこのような問題への解決策として、フローチャートやグラフなどといった概念図が古くから使われてきた。しかし、概念図が文書のように詳細な意味を表現することは難しい。そのため本研究では文書と概念図を関連づけた複合メディアを用いる。我々はその一つの手法として、文書の内部構造から概念図を自動生成する手法を提案してきた[1][2]。ここでは、次のような枠組みを用いている。

- 文書の典型的な意味的構造をマークアップす

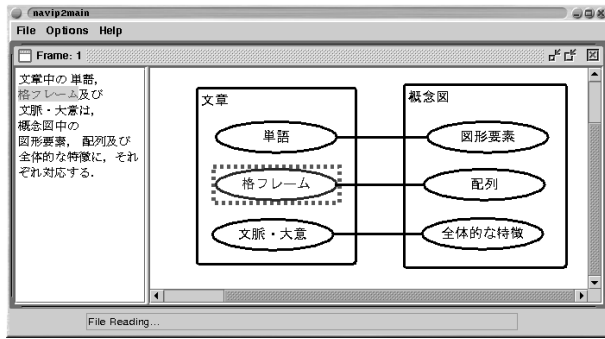


図 1: 簡単な文章に対応する概念図の生成例

る XML タグを定義する.

- タグ付き文書とリンクした概念図を生成する.
- 図に対する編集操作や文書のナビゲーションを行う GUI を提供する.

簡単な図生成の例を図 1 に示す. 図の左側はタグ付き文書からタグを抜いた部分を表示しており, 左側がタグ付き文書を解釈して生成した概念図である. 文書と概念図の要素同士がそれぞれリンクしているため, 図中で「格フレーム」という図形を指示すると, 文書中で対応する部分が強調される. これらの大まかな枠組みについては先行論文 [1][2] で述べているが, 入力されるタグ付き文書から生成される概念図は常に同一であり, ユーザの状況等に応じて適切な表現を選ぶことができなかつた. そこで我々は複数の図的表現を用意し, 概念図生成のさいに選択できるような対応関係を設定した.

以下では, まず文書の意味的構造を分類し, それを記述するためのタグセットについて述べる. 次に, タグと概念図の多対多の対応関係を設定し, その関係を用いて概念図を適応的に生成するプロセスについて述べる.

## 2 文書の意味的構造の記述

図??に示すように, 文書は語句の集合であり, それらの間には多様な関係がある. 文書の意味的構造は, 意味的要素である語句と, この関係によって構成される. 本章ではこのような意味的構造を記述するために, 語句と関係のそれぞれをマークアップしてタグ付き文書として記述する手法について述べる.

表 1: 文書の意味的構造を記述するためのタグセット

タグ名	説明, 属性
<node>, <n>	要素 (語句, 文など) をマークアップする. id=要素に識別子を与える. role=関係を構成する要素の時, 要素に役割を与える. nref=他要素の識別子を参照.
<relation>	関係をマークアップする. structure=関係の構造型. semantics=関係の意味属性.

### 2.1 XML に基づく意味的構造のマークアップ

上述した文書の意味的構造を XML のタグセットとして記述する. XML 自体はツリー型のデータ構造であるが, 要素の属性値として他要素への参照を記述することで, ネットワーク型の構造を定義できる. 意味的構造をマークアップするために定義したタグセットを表 1 に挙げる. 語句や文, 段落, 文章などといった文書中の要素は全て<node>, もしくは短縮形の<n>タグで記述する. この要素タグは, 文書中の連続した部分 (content) に対して識別子や属性値を記述することができる. 要素間の関係については<relation>タグでマークアップし, 後述する構造型を示す structure 属性と意味属性を示す semantics 属性により関係の種類が記述される. <relation>タグで括られる内部には, この関係を構成する要素が<node>タグにより記述される. ここで出てくる<node>タグは上記の要素タグと同じものであるが, 要素の実体を定義せず nref 属性を用いて他の意味要素を参照することもできる. また, 関係に対してどのような役割を持つか role 属性により各要素に記述する. 本手法で用いられるタグは 2 つだけであり, これらのタグに付けられる属性によって多様な意味的構造を記述することができる.

### 2.2 関係の構造型

文書に存在する意味的構造を, 主に代数学的な特性から分類したものがこの構造型である. この

表 2: 関係を構造型で分類

value	典型的な関係 Type of relation
	構造型が取りうる役割要素
	図的表現例
order	順序, 時系列, 因果関係, 階層関係, 包含関係, 上下関係, 等
	upper, medium, lower
equivalence equiv	同値, 並列, 等価, 等.
	obj(object) eq(or neq)
modification mod	説明, 属性付与, 等.
	modifier
	obj(object) modifier
other	その他の関係
	obj(object)

分類には表 2 にあげるように 4 つの型が存在する。order は上位要素と下位要素を持つ順序関係のことで、反対称律<sup>1</sup>および推移律<sup>2</sup>の条件を満たすことが一般的に知られている。これは時系列や序列, 過程, 階層関係など, よく見られる関係であり, 文章はこの構造を中心に展開することが多い。equivalence は equiv と同略記され, 同値関係を示す。これは対称律<sup>3</sup>が成り立つ関係であり, 同位, 等価, 同格などの関係がこれにあたる。

modification は mod と略記し, 他の要素に何らかの情報を付与するような修飾関係である。図的表現例を見ると順序関係に似ているが, 推移律が成り立たない点が大きく違う。例えば, 「青い目の少女」という表現があった場合, 「青い」は「目」を修飾し, 同様に「目」は「少女」を修飾するが, 「青い」は「少女」を修飾したことにはならない。また, other はこれらに分類されない関係を意味する。

<sup>1</sup>  $a, b$  を要素,  $R$  を関係としたときに,  $aRb, bRa \Rightarrow a = b$

<sup>2</sup> 同様に,  $aRb, bRc \Rightarrow aRc$

<sup>3</sup> 関係が対称であること。  $aRb \Rightarrow bRa$

表 3: 関係を意味属性で分類

属性値	関係例.
time	時系列, 推移, 順序, 等
cause	因果関係, 理由と結果, 等
space	空間的位置関係.
in/out	入出力. 原材料や製品・中間生成物との関係, 等
process	過程, 物の流れ, 等
set	組織, 集合論, 上下関係, 等.
term	語義, 等.
other	その他の関係

私たちが普段使っている

```

<node id="diagram">概念図</node>
には主に 2 つの種類があります.
<node id="connect">連結図</node>
と
<node id="area">領域図</node>
です.
<relation structure="order"
  semantics="set">
  <node role="upper" nref="#diagram" />
  <node role="lower" nref="#connect" />
  <node role="lower" nref="#area" />
</relation>

```

図 2: タグ付き文書の例

### 2.3 関係の意味属性

前章の構造型だけでは, 多様な関係を記述しきれない。そこで, 意味的な特性で関係を分類したものがこの意味属性である。関係が何を意味しているのか記述することで, 意味的な検索や知的な情報フィルタリングが可能になる。

このような意味属性は多くの種類が考えられるが, その中の主なものを表 3 にあげる。time 属性は時間を示すが, order 構造型と組み合わせることで, 時系列という意味の関係を記述できる。cause は, 因果関係を記述するための属性である。また space は物理的な位置関係を示す。process と in/out は, 過程とその前後の入出力物を意味するが, 前者が

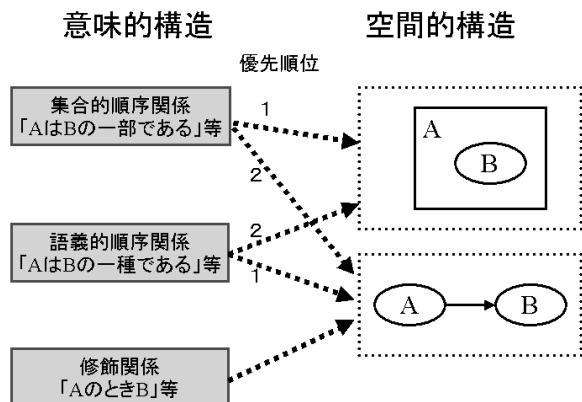


図 3: 意味的構造と空間的構造の対応

流れを中心にとらえた意味であるのと対照的に後者はモノを中心とする意味である。set は集合的な関係に付けられる属性であり、組織や集合の階層関係、包含関係を意味する。term は語義関係を表わすが、例えば「ペンギンは鳥の一種である」などといった上下関係を示している。

## 2.4 タグ付き文書の例

簡単な文章にタグを付加した例が図2である。前半が文章自体であるが、ここでは要素タグが語句を定義している。2行目は「概念図」という語句に diagram という識別子を与えている。同様に4行目は「連結図」に connect という識別子を、6行目は「領域図」に area という識別子を与えている。後半は関係の記述であり、構造型が order かつ意味属性が set という組み合わせの構造を記述している。また、識別子 diagram で参照される「概念図」に役割 upper を、connect 及び area で参照される「連結図」「領域図」に役割 lower を設定している。この結果、図2では、「概念図」の内部に「連結図」「領域図」が存在するという集合的な順序関係を記述している。

## 3 意味的構造と概念図の対応

### 3.1 空間的構造

空間的構造は、図4のように2つもしくはそれ以上の図形からなり、それら図形の間には接触・隣

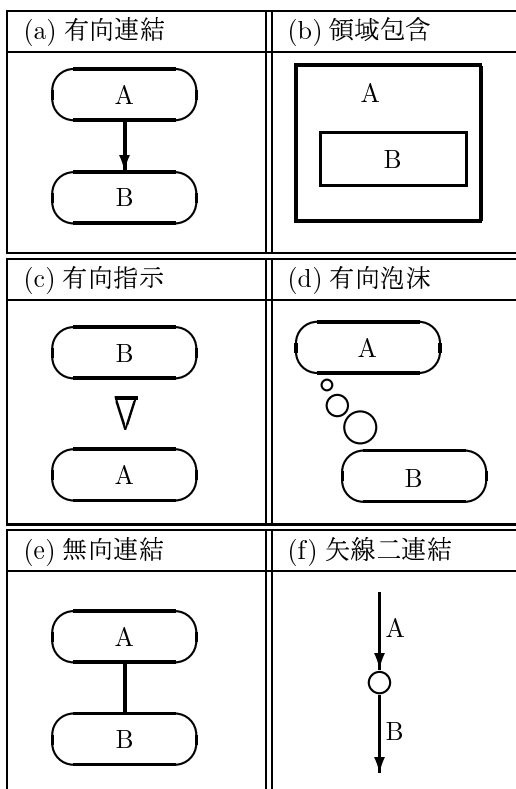


図 4: 空間的構造の例

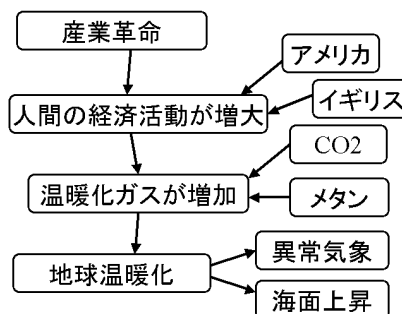


図 5: 内容が分かりにくい図: 地球温暖化について述べられた文章を矢線のみで表現した。

接などの空間的位置関係が定義されている。例えば、図4(a)の有向連結の場合、必要な要素は「A」と「B」と矢線である。またAと矢線の根本、Bと矢線の先端の間には接触関係が存在し、また全ての関係の間には垂直に整列するという関係が存在する。このように基本的な空間的關係を組み合わせることで、概念図を構成することができる。

関係の種類	第一候補	第二候補	第三候補
order & time	(a) 	(c) 	(e) 
order & process	(a) 	(c) 	(f) 
order & inout	(a) 	(f) 	(c) 
equiv & time	(e) 	none	none
mod & term	(d) 	(a) 	(b) 
order & set	(b) 	(a) 	(e) 

図 6: 順位付き対応関係の例

### 3.2 意味的構造と空間的構造の対応

本研究の従来手法では、意味的構造と空間的構造の間に一対一の対応関係を定義していた。しかし、多対多の対応関係を設定する場合には、複数の意味的構造を同じ空間的構造で表現する可能性があり、ユーザの誤解を招く恐れがある。例えば図5では、いくつかの種類の関係が同じ矢線で示されており、それぞれの関係を見分けることが難しく、表現されている内容がわかりにくい。そこで、同じ空間的構造が異なる意味的構造に割り当てられることが無いように、対応テーブルに工夫を施す必要がある。

この問題に対して、図6のように、順位がつい



図 7: 図生成例: order & time 関係を有向矢線で表現した概念図

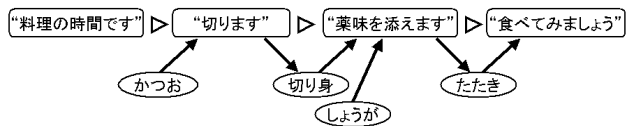


図 8: 図生成例: order & inout 関係が混在するため、order & time 関係を有向指示で表現した概念図

た対応テーブルを設定した。例えば、時間の順序関係 (order & time の組み合わせ) の場合、他に支障無い限り空間的構造 (a)(有向連結) が割り当てられる。しかし他に優先すべき関係が既に (a) に割り当てられていた場合、競合を避けて空間的構造 (c)(有向指示) に割り当てられる。さらに (c) も競合する場合は、(e)(無向連結) に割り当てられる。このように競合を避けることができるよう、順位付きの多対多の対応関係を設定した。

実際に割り当ての変更が行われた例を図7および図8に示す。どちらの図も、入力したタグ付き文書に時間の順序関係が存在している。図7では文書中に関係は一つしか存在しないため、時間の順序関係には第一候補の空間的構造 (a) が割り当てられる。しかし図8では、「入出力の順序関係」(order & inout の組み合わせ) が同時に存在する。また、タグつき文書を入力すると同時に、時間の順序関係よりも入出力の順序関係を優先すべきとユーザが指示しているものとする。2つの関係はどちらも (a) を第一候補としているが、これでは競合が起こるためどちらかが別の空間的構造を選択せねばならない。しかし前述のように関係の優先順位が与えられているため、まず入出力の順序関係が第一候補の (a) に割り当てられ、次に時間の順序関係が第二候補の (c) に割り当てられる。その結果、図8では入出力が矢線によって示された概念図となる。それぞれの過程にどのような要素が入力し出力されているかわかる。また時間の順序関係は矢尻の軌跡によって指示された概念図となった。

このような多対多の対応関係を用いることで、ユーザの好みに合わせて図表現を変更することができる。例えば、ユーザが手順の説明を求めている

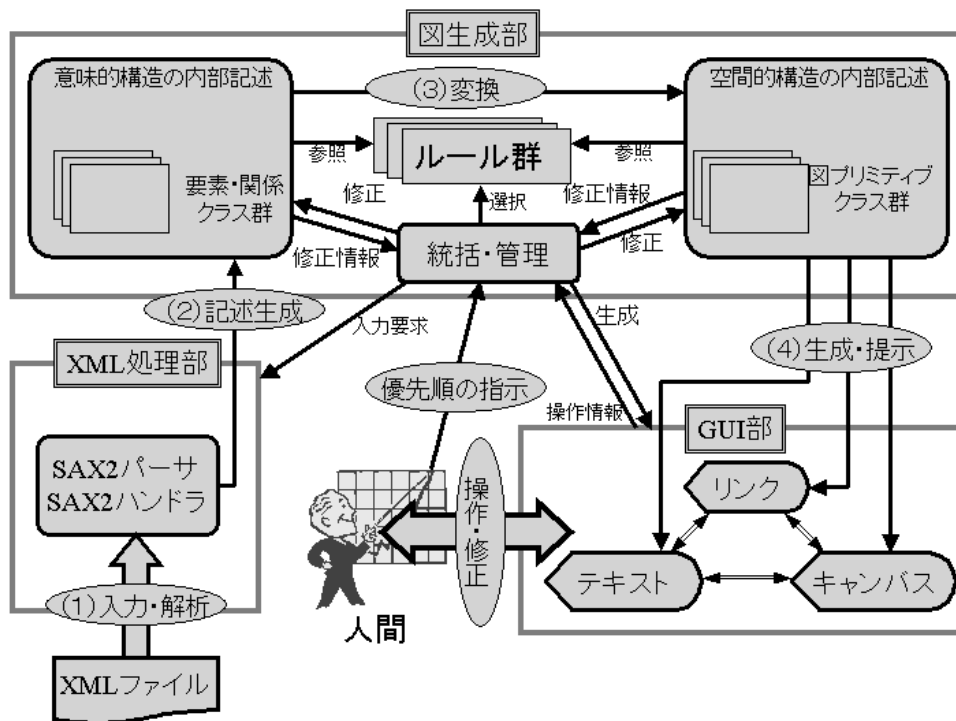


図 9: 図生成システムの構成

なら，入出力の順序関係を最優先することで手順が見やすい概念図を生成することができる。将来，このように単純な割り当て変更アルゴリズムではなく，全体のコストを最小にするような割り当てアルゴリズムが望まれる。しかし，具体的なコストの評価方法などを定めていないため，これは今後の研究課題である。

## 4 概念図の生成手法

本稿では筆者らの従来研究 [1] で使用していた概念図生成システムを元に新たに図生成システムを開発した。新システムでは，次に示す従来システムの大きな枠組みをそのまま用いているが，各処理は完全に再設計している。

- (1) 文書中の意味的構造を入力・解釈
- (2) 解釈結果から意味的構造の内部記述を生成
- (3) 対応関係から空間的構造の内部記述を生成
- (4) 空間的構造を表現する図を生成

このような多段階処理をしていくことで，最終的な概念図を得る。詳細な処理の流れを図 3.2 に示す。

図中の括弧付き番号は，上で示した処理の流れに対応している。

意味的構造と空間的構造の間の対応関係の設定は，あらかじめ設定したルール群を参照することで行う。各意味的構造の優先順は外部から与え，適切な対応関係が設定されるよう調整する。次章では，この概念図生成システムを用いて行った実験について述べる。

## 5 実験

まず，図生成システムの基本的機能を確認するため簡単な概念図を生成する実験を行った。図 10 のタグ付き文書をシステムに入力したところ，図 11 を得た<sup>4</sup>。語句の階層関係が囲みで示され，また修飾の関係が矢線で表現されている。

次に，多種類の意味的構造を含む複雑な概念図の生成を試みた。ここで用いたタグ付き文書は長いため，図 12 に示すのは一部を抜粋したものである。左半分はおもちゃの組立て過程を順序立てて示しており，完成までの流れが上から下へ示されて

<sup>4</sup>ただし，図描画アルゴリズムの未実装によりレイアウトが崩れていたため，手動で編集操作を施した。以下同様。

```

<n id="sentence">文章</n>が
<n id="word">単語</n>や
<n id="phrase">句</n>といった
<n id="ele">要素</n>と、それらの要素間の
<n id="rel">関係</n>により構成されていると考えます。

関係には、例えば
<n id="shift">推移</n>や
<n id="procs">過程</n>、
<n id="hier">階層</n>などがあります。
これらの要素と関係を
<n id="tag">タグ</n>を使って表すことにより、
<n id="txt">タグ付き文書</n>を作ることができます。

<relation structure="mod" semantics="content">
  <n role="modifier" nref="#tag" />
  <n role="object" nref="#ele" />
</relation>
...
<relation structure="mod" semantics="term">
  <n role="modifier" nref="#hier" />
  <n role="object" nref="#rel" />
</relation>
<relation structure="order" semantics="set">
  <n role="upper" nref="#txt" />
  <n role="lower" nref="#sentence" />
  <n role="lower" nref="#rel" />
  <n role="lower" nref="#ele" />
  <n role="lower" nref="#tag" />
</relation>

```

図 10: 入力したタグ付き文書 (抜粋)

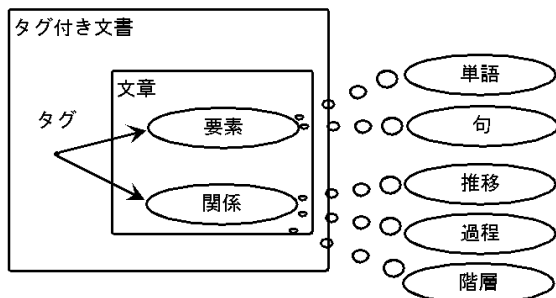


図 11: 簡単な概念図の生成結果

いる。右半分は組立てに用いられる部品を羅列してあり、どの部品が左側のどのプロセスに入力されるかが矢線で示されている。加えて、「トイカー」に一つ、「パワーツール」に3つの特性の修飾要素が係っているのがわかる。また「トイカー」に係っている修飾要素「前後に走らせることができます」は、さらに過程の修飾関係によって修飾されている。このように過程の順序は矢尻の軌跡、入出力の流れは矢線、そして過程と特性の修飾関係はダブル状の吹き出しと囲みで行われているが、集合

```

それでは組み立ててみます。
<n id="B1">シャーシ</n>に
<n id="X3"><n id="B2">タイヤ</n>を取り付け</n>
ます。
まず、<n id="A1">前輪を取り付け</n>ます。
次に<n id="A2">後輪を取り付け</n>ます。
そして、
<n id="A3"><n id="B3">ボディ</n>を取り付け</n>
ます。
...
<relation structure="order"
  semantics="process">
  <n role="upper" nref="#A1" />
  <n role="lower" nref="#A2" />
</relation>
<relation structure="order"
  semantics="inout">
  <n role="input" nref="#B1" />
  <n role="object" nref="#X3" />
</relation>
<relation structure="order"
  semantics="inout">
  <n role="input" nref="#B2" />
  <n role="object" nref="#X3" />
</relation>
<relation structure="order"
  semantics="inout">
  <n role="input" nref="#B3" />
  <n role="object" nref="#A3" />
</relation>
<relation structure="order"
  semantics="process">
  <n role="upper" nref="#X3" />
  <n role="lower" nref="#A3" />
</relation>
<relation structure="order"
  semantics="set">
  <n role="upper" nref="#X3" />
  <n role="lower" nref="#A1" />
  <n role="lower" nref="#A2" />
</relation>
...

```

図 12: タグ付き文書 (抜粋) : おもちゃの組立て

の順序関係が他の関係との競合により表現されておらず、過程と入出力の流れが混雑した見にくい図になってしまっている。

そこで、この文書を入力する際に、集合の順序関係を優先するようシステムに指示した。図 13 では、過程の修飾関係が最優先されて囲みに割り当てられていた。しかし、優先順を変更したことにより、集合の順序には第一候補の囲みが割り当てられ、また囲みを割り当てられなかった過程の修飾関係は他の関係との競合により図上から削除された。

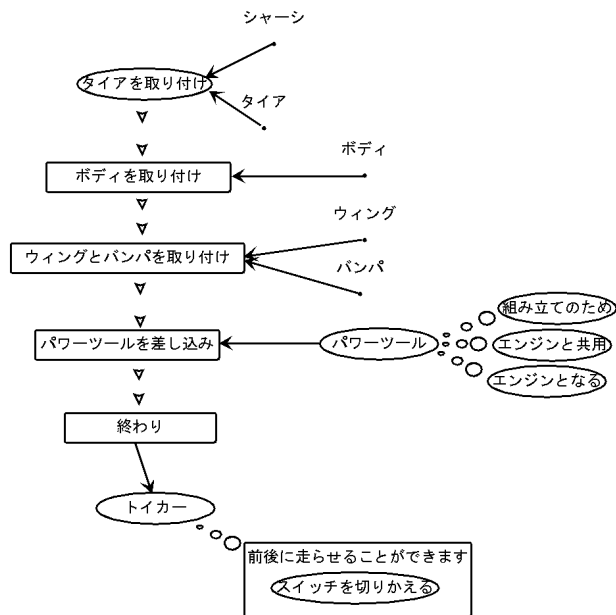


図 13: 多種の関係が入り組んだ概念図の生成結果

その結果得られた概念図が図 14 である。過程と出力要素の各集合が、それぞれ「トイカーの組み立て方」「トイカーのパーツ」という 2 つの囲み要素で大きくくくられることで、他集合と混同しないよう分割されている。

このように、タグ付き文書のどのような意味的構造に着目するかで概念図の表現を選択することができ、重要な関係に最適な図的表現を行うことができた。

## 6 おわりに

本稿では、タグによって文書の意味的構造を記述し、概念図を生成する手法について提案した。文書中の意味的構造と概念図の空間的構造を分類し、これらに多対多の対応関係を設定した。その対応関係を利用し、概念図を生成するシステムを設計・実装した。図生成システムにより、ユーザの要求に応じて重要な関係が最適な空間的構造として表現される概念図を生成できることが確かめられた。

今後、意味的構造と空間的構造との間に設定した対応関係について検証を重ねていく予定である。そのために、対応関係の良さをアンケート等を行って評価している途中であり、その結果に基づいてシステムの実装を進める予定である。

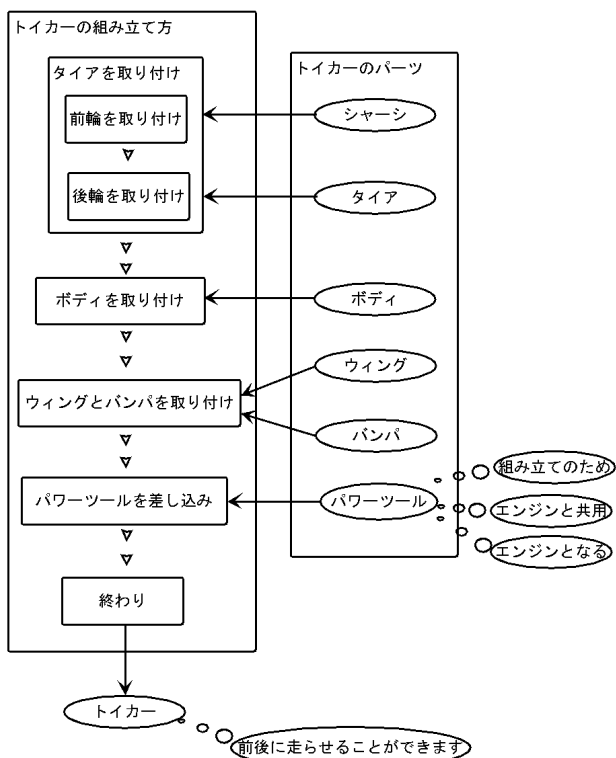


図 14: 意味的構造の優先順位を変更した例：処理の推移が矢尻の軌跡により示され、入出力が矢線で示されている。

## 参考文献

- [1] Murayama, M., Nakamura, Y. and Ohta, Y.: Diagram Generation From Tagged Texts Toward Document Navigation, in *IEEE ICME 2001 proceedings FP0.04* (2001).
- [2] 村山正司, 中村裕一, 大田友一: 概念図の自動生成による文書内容の可視化-タグ付き文書からの自動変換-, 第 5 回知能情報メディアシンポジウム, 電子情報通信学会 (1999).
- [3] 橋田浩一: GDA: 意味的修飾に基づく多用途の知的コンテンツ, *人工知能学会誌*, Vol. 13, No. 4, pp. 528-535 (1998).