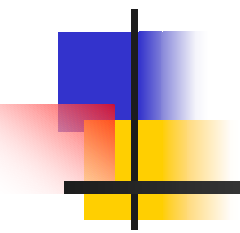


ソフトウェアパターンの自動的な体系化の試み

久保淳人　鷲崎弘宜　深澤良彰
早稲田大学理工学部

背景と目的

- WWWには多数のソフトウェアパターン文書が存在する
  - 体系化されていれば検索や利用に便利
  - それぞれのサイトにおいては体系化されている
  - しかし，他サイトのパターンとの関連は希薄
- 網羅的な体系化における問題
  - 規模：人手での網羅的な体系化は不可能に近い
  - 人間が気付かないパターン間の関連
    - そもそもパターンの存在を知らない（全てを把握しきれない）
    - 存在を知っていても気付くことができない関連
      - 粒度/適用分野/表現方法等の微妙な相異

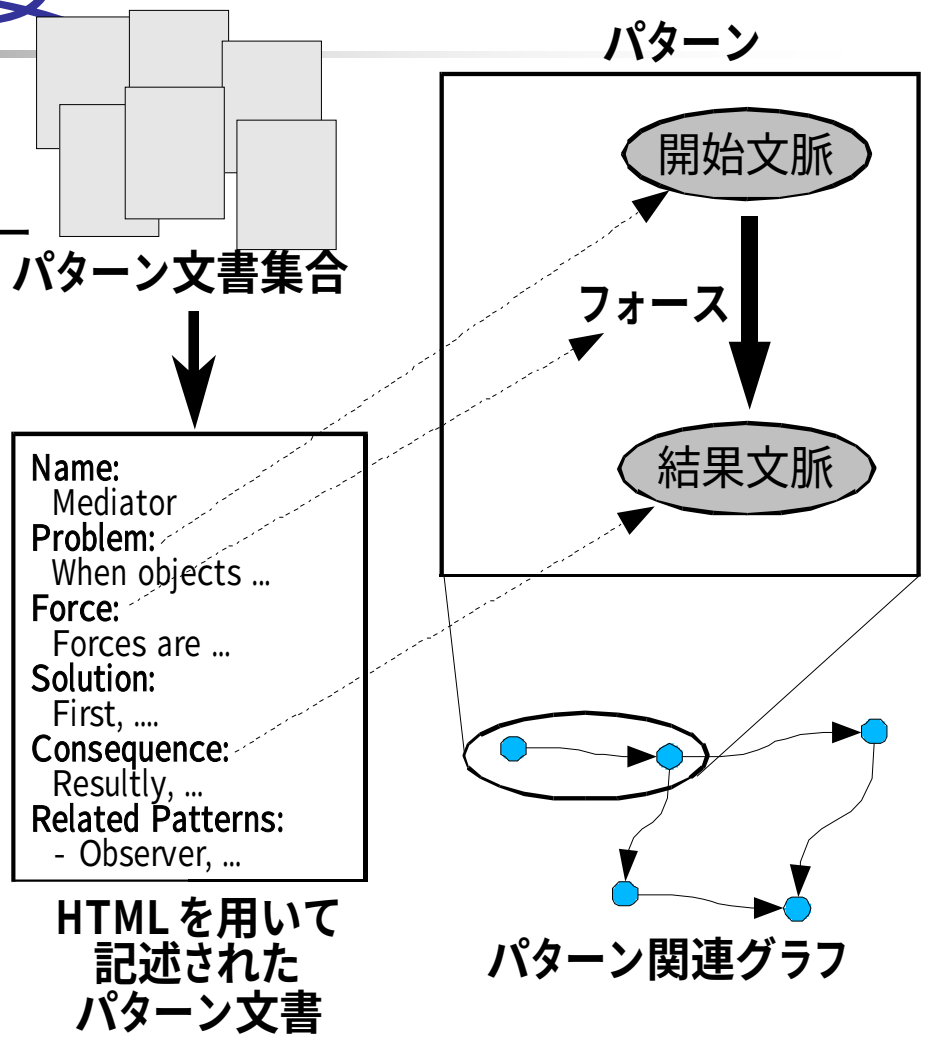
提案システム

自動的にパターン文書を収集して，パターン間の関連を抽出するシステムを提案

- パターンの自動的，網羅的な体系化
- 人間が気付かなかったパターン間の関連を示唆
  →人間による体系化と相補的関係に？

システムの概要

- 入力
  - HTMLを用いて記述されたパターン文書集合
- 処理
  1) HTML文書の解析
  2) パターンフォームの判定
  3) パターン間の関連の分析



1) HTML文書の解析

- 入力されたHTML文書の，パターン文書としての構造を取得したい
- 問題
  - HTML文書は，パターン文書としての明示的な構造を持たない
- 解決案
  - 文書片に分解する
  - 文書片の並びは，見出しとそれに続く本文の繰り返しと仮定
  - 見出しを特徴づけるタグ，本文を特徴づけるタグ

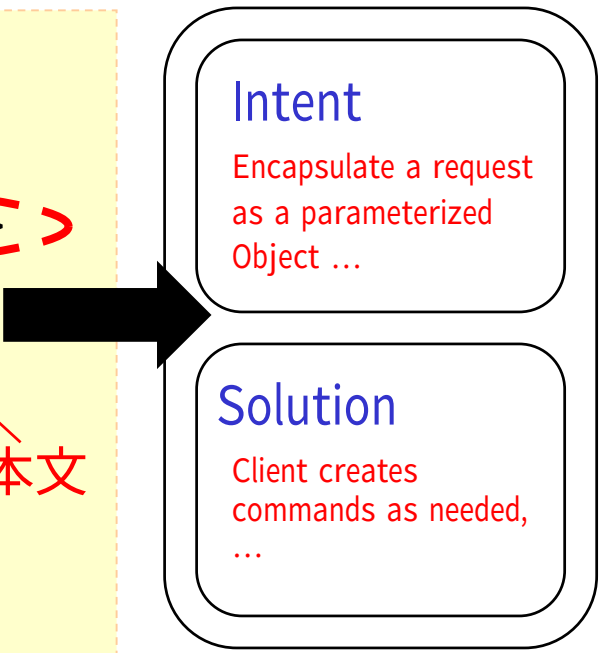
HTML文書の解析例

例：Command パターン

見出し

```
<h2>Intent</h2>
<p>Encapsulate a request as a parameterized object ...</p>
<h2>Solution</h2>
<ol>
<li>Client creates commands as needed, ...</li>
<li>Each SpecificCommand is ...</li>
</ol>
```

本文

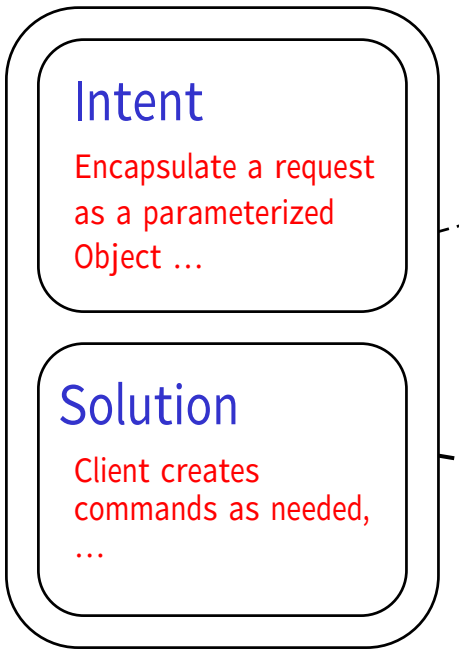


「h1～h6タグに囲まれた文書片は見出し」「pタグやliタグに囲まれた文書片は本文」というルールで解析

2) パターンフォームの判定

- パターン文書が，どのようなフォームに沿って書かれているのか知りたい
- 解決案
  - よく知られたパターンフォーム (GoF，Coplien等) について，判定のための情報を用意
    - 例えば…「Solutionという単語を含む見出しが存在する」
  - それぞれのパターンフォームとの合致度を得る
  - 最も合致するフォームをそのパターン文書のパターンフォームと見なす

パターンフォーム判定の例



GoF Form

(条件1) Intentという語を含む見出しがある

(条件2) Solutionという語を含む見出しが…

条件8個中，6個を満たした → 合致度0.75

Coplien Form

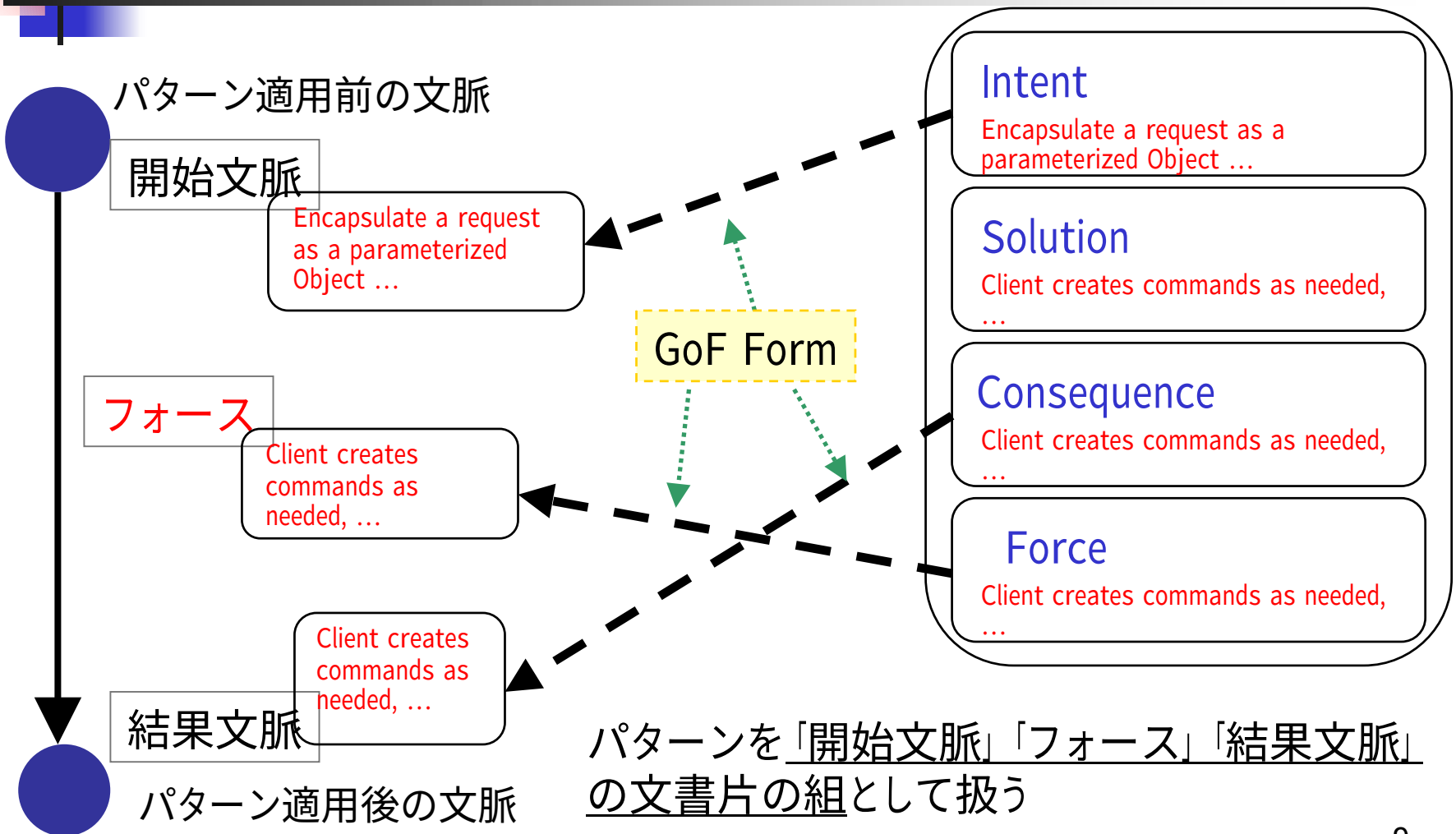
(条件1) Forceという語を含む見出しがある

(条件2) Problemという語を含む見出しが…

条件10個中，4個を満たした → 合致度0.4

GoF FormとPoSAフォームについて合致度を求めた結果，GoF Formの合致度が高かった→このパターン文書はGoF Formであると見なす

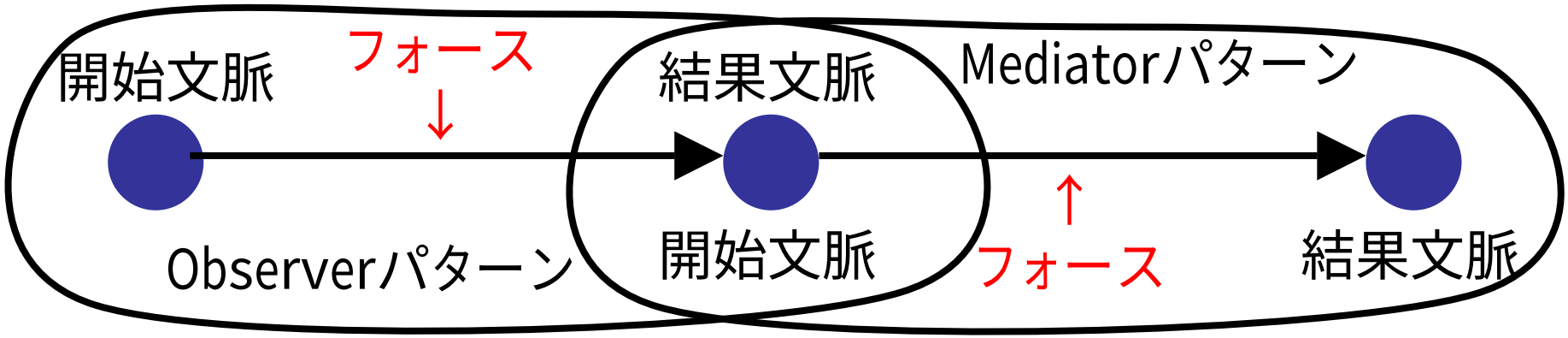
パターン文書からパターンを得る



パターンを「開始文脈」「フォース」「結果文脈」の文書片の組として扱う

3) パターン間の関連を分析

- パターンを適用した結果として別の文脈を得る[1]
  - → 得た文脈は他のパターンの開始文脈
  - → さらにパターン適用可能？
- 類似
  - 開始文脈同士
  - 結果文脈同士
  - 結果文脈と開始文脈（例↓）
  - フォース同士
- 類似した文脈を同一視することでグラフを作る



[1] 鷲崎 弘宜, 深澤 良彰, ソフトウェアパターン研究の現在と未来, 情報処理学会第141回ソフトウェア工学研究会 (2003)

パターン関連グラフ(PRG)

- 連続するパターンの適用をラベル付き多重有向グラフとして表現する
  - ノード：文脈
  - エッジ：パターン適用
  - ラベル：フォース

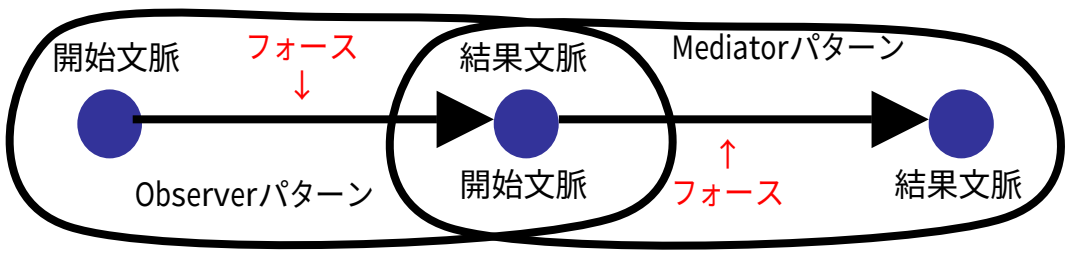
ソフトウェア変更プロセスのモデル化？

パターン集合 $P = \{(c_i, c_j, \lambda_k) \mid c_i, c_j \in C, i \neq j, \lambda_k \in \Lambda\}$

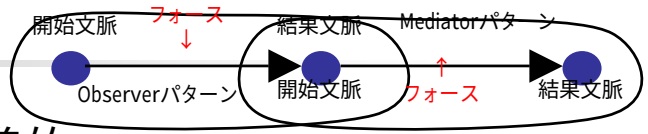
文脈集合 $C = \{c_1, c_2, \ldots, c_n\}$

フォース集合 $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$

パターン関連グラフ $PRG = (C, P, \Lambda)$



類似度の算出手法



- ベクトル空間モデル；情報検索においては標準的
- 文書片に含まれる各単語の重みベクトル
  - 各単語の重みは（標準的手法である）tf*idf法で算出
- 2つの文書片の類似度は，両ベクトルの余弦

開始文脈

Encapsulate a request as a parameterized Object …

(0.5, 0.2, 1.2, …) $\vec{a}$

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|}$$

フォース

$\vec{b}$ (0.6, 0.4, 0.8, …)

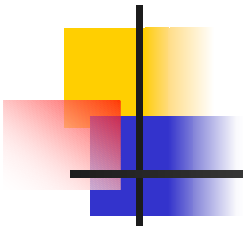
開始文脈

結果文脈

Client creates commands as needed, …

結果文脈

例：開始文脈と結果文脈の類似度計算



実験

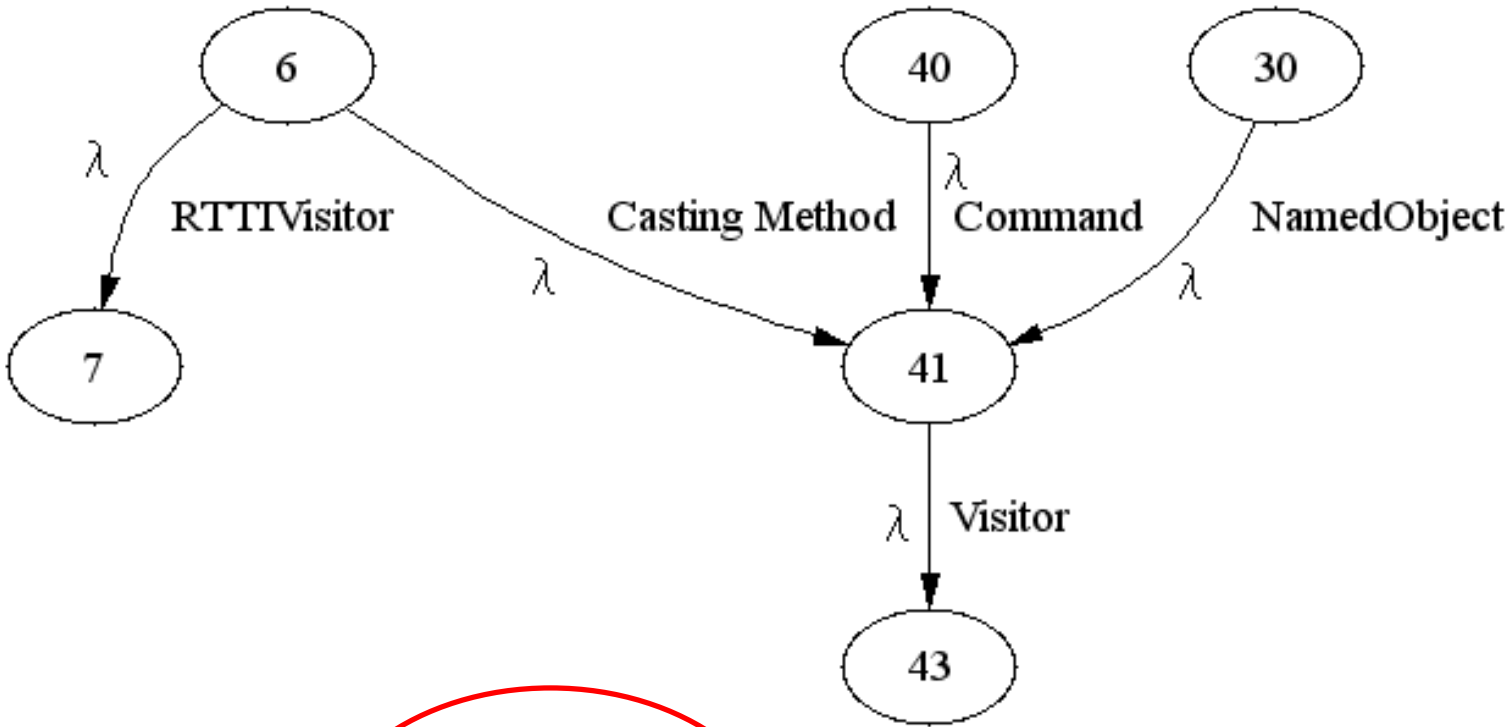
- The Object Oriented Patterns Digest (http://patterndigest.com)のパターン文書42個を入力
- 下記の関連の上位15個(計45個)について，抽出されたパターン間の関連と，パターン文書中に明示されている関連とを比較
  - 開始文脈同士の類似
  - 結果文脈同士の類似
  - 結果文脈と開始文脈の類似
- パターン関連グラフを作成

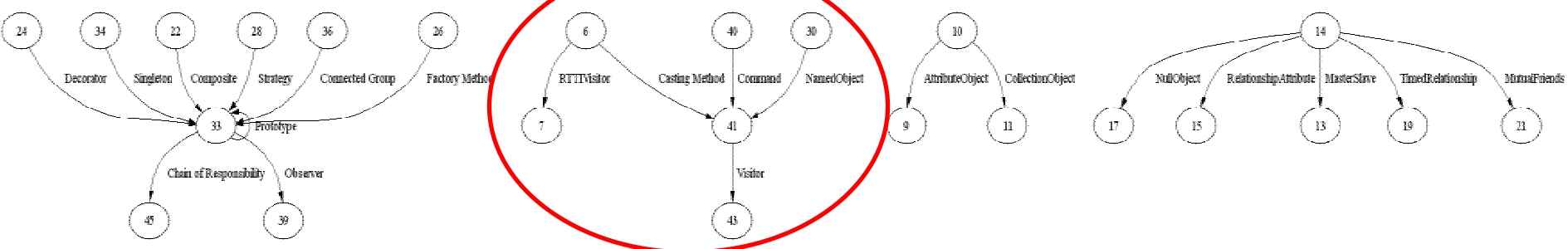
結果と考察

- 結果
  - 再現率（漏れがないか） 0.237：低い
  - 精度（ゴミがないか） 0.311：低い
  - フォールアウト（不要な関連を排除する能力） 0.0387：良好
- 考察
  - 再現率の低さ：サンプル数が少なすぎた？
  - サンプル数によって結果が変わる；最適な閾値は？
  - （汎用の情報検索と異なり）精度の低さ＝失敗とは限らない
    →明示されていない関連の発見につながる可能性

全体：$_{42}C_2 = 861$	発見した関連	発見しなかった関連
明示された関連	14	45
明示されていない関連	31	771

パターン関連グラフ


6
40
30
λ
RTTIVisitor
λ
Casting Method
λ
Command
λ
NamedObject
7
41
λ
Visitor
43


24
34
22
28
36
26
Decorator
Singleton
Composite
Strategy
Connected Group
Factory Method
33
Prototype
Chain of Responsibility
Observer
45
39
6
40
30
RTTIVisitor
Casting Method
Command
NamedObject
7
41
Visitor
43
10
AttributeObject
CollectionObject
9
11
14
NullObject
RelationshipAttribute
MasterSlave
TimedRelationship
MutualFriends
17
15
13
19
21

まとめと今後の課題

- まとめ
  - パターン間の関連の分析の自動化を試みた
  - HTML文書からパターンを抽出して，パターン間の関連を分析した
- 課題
  - 条件を変えて評価
    - サンプル数を変更したときの精度･再現率の変化
  - 複数の作者のパターン文書を入力したときの評価
    - ここが主眼
    - 異なるサイトのパターン間でも関連を発見できるか？

性能評価の指標

- 再現率：結果の漏れがないか

  R = w / (w + x) (高いほど良い)

- 精度：結果にゴミがないか

  P = w / (w + y) (高いほど良い)

- フォールアウト：不要な関連を排除する能力

  F = y / (y + z) (低いほど良い)

	発見した関連	発見しなかった関連
明示された関連	w	x
明示されていない関連	y	z

tf*idf法

- ある文書中における単語重みの算出手法
  - 情報検索の基本的手法
- tf: Term Frequency
  - ある文書における，ある単語の出現頻度
  - 例：文書dに単語tが5回現れる場合，

    $$tf(d, t) = 5$$
- idf: Inverse Document Frequency
  - ある単語を含む文書の特定性
  - N: 全文書数
  - df(t): 単語tが現れる文書の数
  - $idf(t) = \log(N / df(t)) + 1$
- tf*idf
  - 文書dにおける単語tの重み $w(d, t) = tf(d, t) * idf(t)$