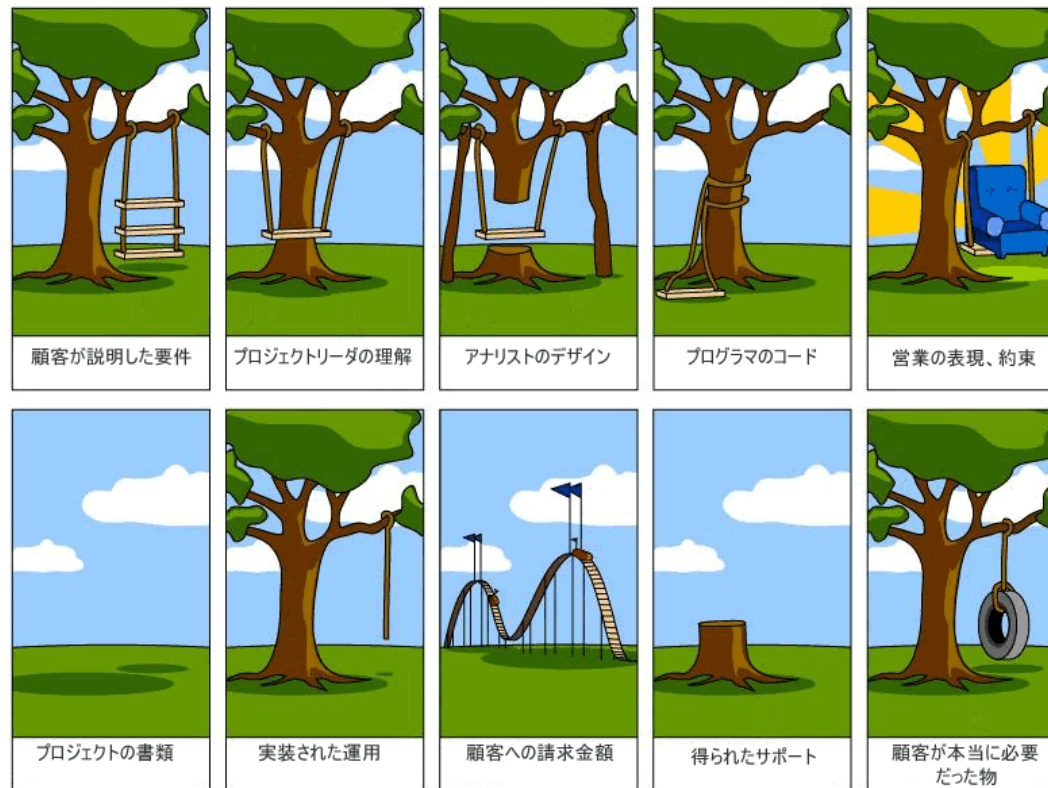


# 研究内容紹介：要求工学

- ▶ 要求工学の重要性: StandishのCHAOSレポート
  - ▶ プロジェクト失敗の原因：
    - ▶ 2位「不完全な要求と仕様」，3位「要求と仕様の変更」

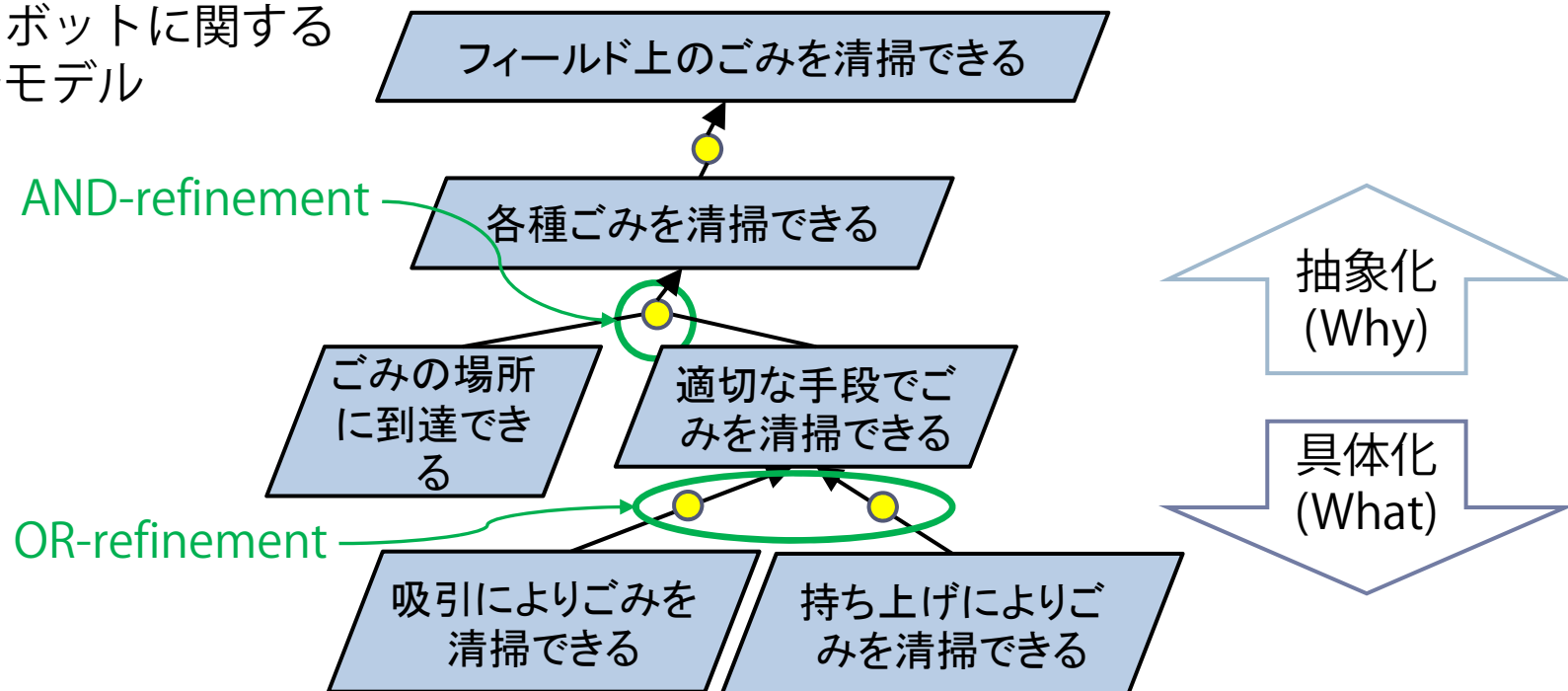


@ITより引用  
原著: University of  
London Computer  
Center Newsletter,  
No.53, March 1973

# ゴールモデル

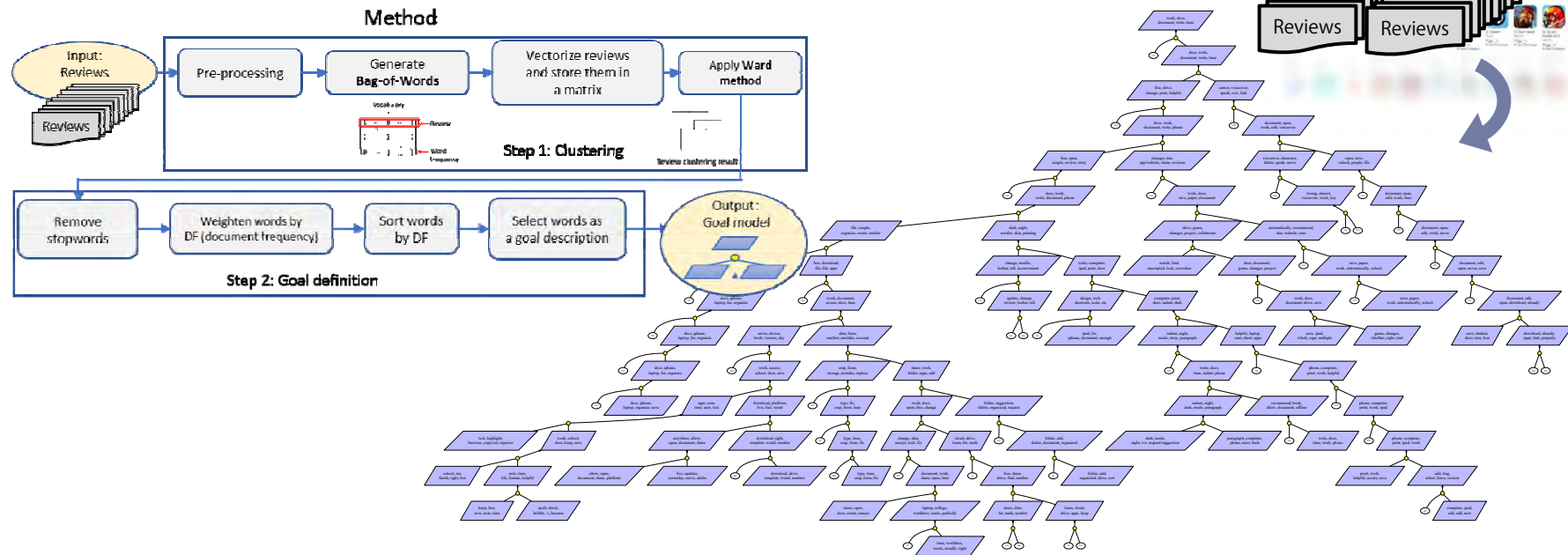
- ▶ ソフトウェアが実現すべき**要求 (ゴール)**を**構造化**し、  
目標から手段へと階層的に展開したモデル
  - ▶ 分析漏れや要求間の矛盾発見に効果的

例: 清掃ロボットに関する  
ゴールモデル



# レビューからのゴールモデルの自動生成

- ▶ 膨大なアプリケーションレビューから要求を抽出
  - ▶ 類似レビューを分類し, ゴールモデルを構築
    - ▶ クラスタリングによりゴールを発見し構造化
    - ▶ 自然言語処理技術, Deep Learningを用いてゴールラベリング



# 研究内容紹介：その他 (共同研究など)

- ▶ 可視化ツール (SUNTORY関連会社)
- ▶ 遺伝的プログラミング (鹿島建設)
- ▶ IoTシステム開発手法 (信州大学, 日本ユニシス)

The collage illustrates various research and development activities. On the left, a flowchart details a text analysis process: 'Text case descriptions - screen is displayed' leads to 'screen display', which is then 'Identify' and 'search' to produce 'Tf(7.1,6.5,0,...)' and 'Tj(0.9,5.0,...)'. This is followed by '2. Represent as vectors of row numbers and normalizes using TF-IDF', resulting in vectors like 'A: (70, 7.65, 56, 3,...)', 'B: (0.44, 32, 5, 0,...)', and 'C: (75, 9.55, 63, 1,...)'. Step 3 involves 'Calculate cosine similarity between specifications', yielding values like 'AB: 0.023', 'AC: 0.773', and 'AD: 0.014'. Step 4 is 'Assign all text case descriptions to specifications'. Below this are images of a vineyard and a solar panel. In the center, a 3D network visualization shows nodes and connections. To the right, a code editor window displays code, and below it, a 3D bar chart and a 3D cube visualization. At the bottom right, a complex flowchart diagram shows various interconnected components and processes.

▶ 9 ソフトウェア開発, プログラミングに興味のある方を募集します!