

# 要求分析による オントロジーの活用法

2005年4月22日

海谷 治彦 (信州大)

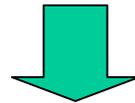
佐伯 元司 (東工大)

# 目次

- 背景と研究目的
- 要求分析で重要な意味処理
- 本研究でのオントロジ
- 矛盾と欠落の検出
- 意味を考慮した要求仕様書のメトリクス
- 変更予測
- まとめと今後の課題

# 研究目的と問題点

- 要求仕様書の機械的な分析支援をしたい.
- 意味を無視した字面の分析では大したことはできない.
- 特殊な記述様式(半形式言語や厳格な形式仕様)の導入はコスト的に割が合わない.
  - 技術者の教育コスト等.



- 低コストな方法で今まで考慮した意味処理を実現したい！ (軽量化意味処理)

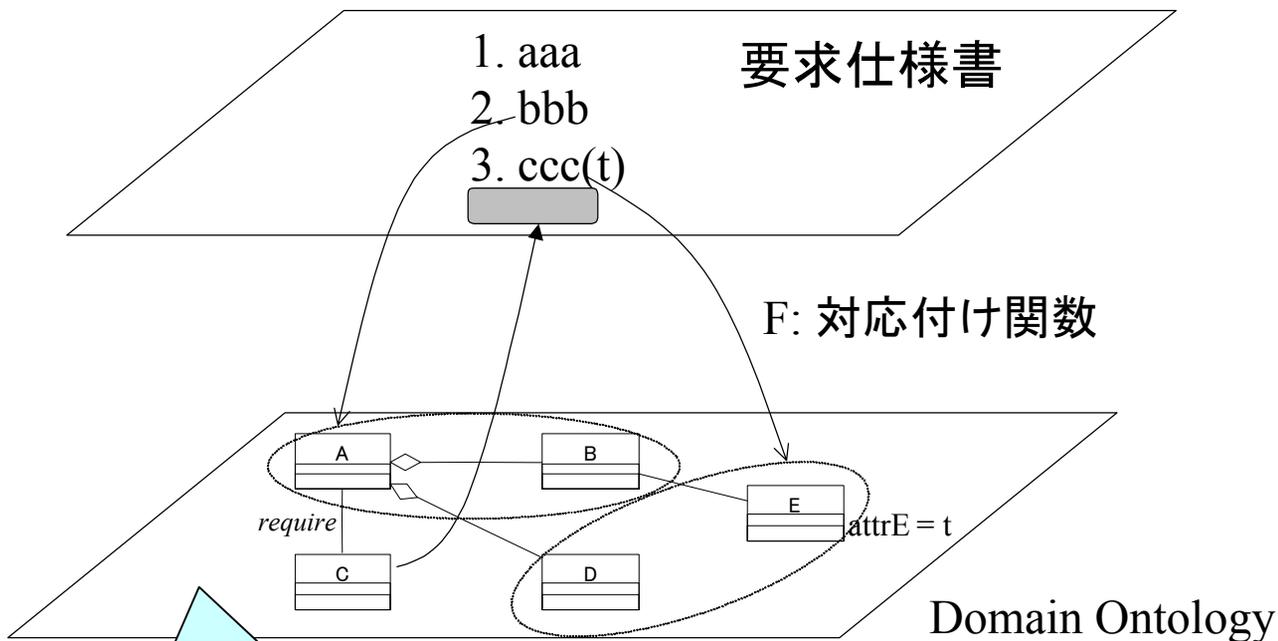
# 要求分析での重要意味処理

- 矛盾の検出
  - 矛盾した記述で設計・実装したら大変.
- 欠落事項の検出
  - 後段での要求追加は回避したい.
- 妥当性の検出
  - 客の要望に合致してないと, そもそも意味無し.
- 非曖昧性の検出
  - 後段での誤解・誤実装の原因となる.

# オントロジーの利用可能性

- 有名な定義  
formal explicit specification of shared conceptualization
- 本研究では,
  - 要求項目の意味付けに使える構造
  - 意味付けをする構造上で計算をする機構が必要である.
- オントロジー ⇒ シソーラス + 推論規則

# 要求仕様書とオントロジの関係



この上で各種の計算(推論)をすることで意味処理を代行する.

# 要求仕様書の構造

- 自然言語で記述されたソフトウェア要求仕様書.
- 箇条書きされた要求項目の列.
  - 将来はIEEE830の推奨書式も考えたい.
- 当面, 図表は対象外.

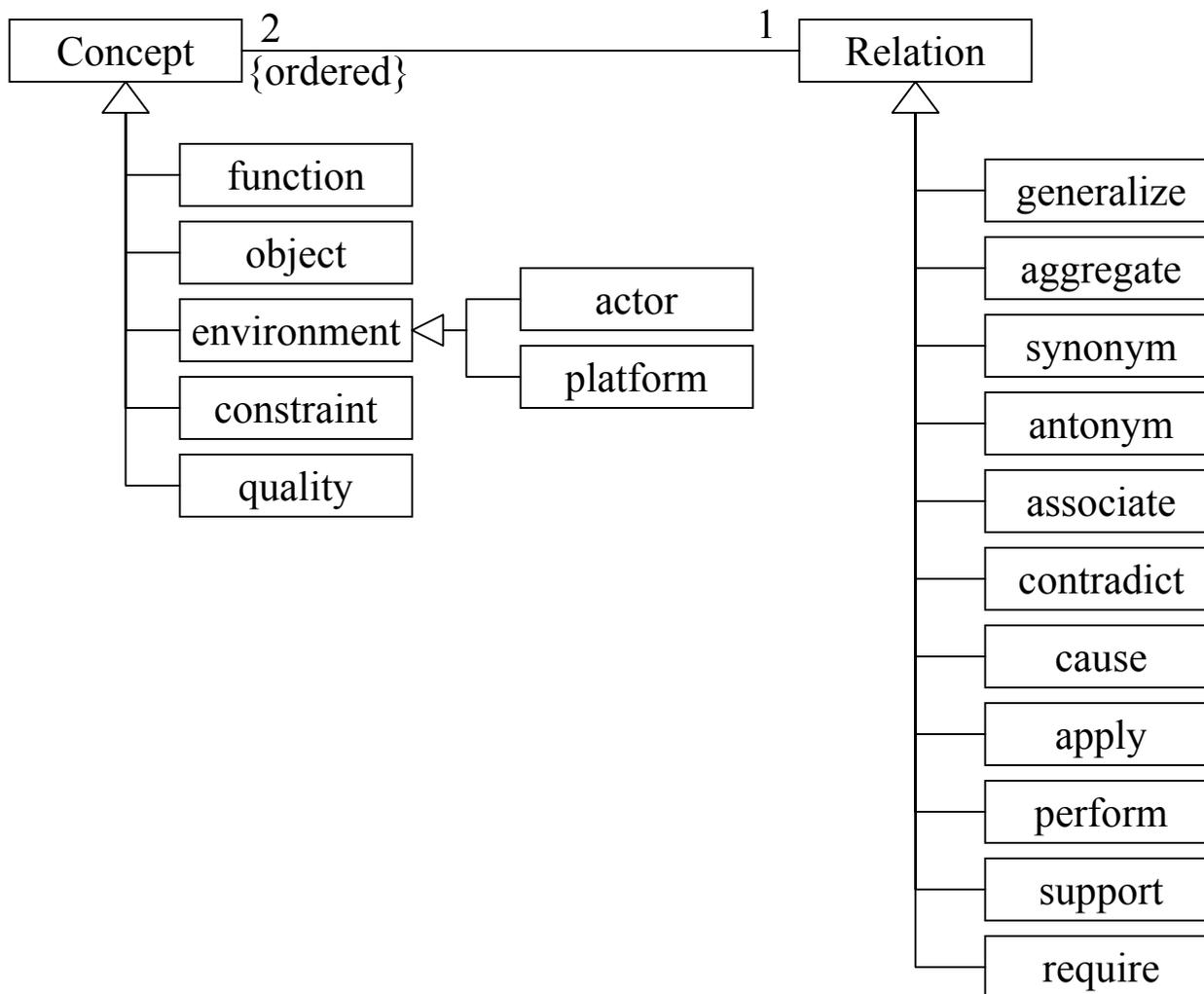
# 例: 音楽プレイヤーの要求

1. 音楽の再生できる.
2. 一時停止できる.
3. 前後曲の頭だしができる.
4. 曲操作は早い.
5. 音楽の早送り, 巻き戻しができる.
6. 音量調整と消音ができる.
7. 曲毎に音量設定ができる.
8. 再生速度を変更できる.
9. 再生音程を変更できる.
10. プレイリストのランダム再生ができる.
11. プレイリストに関して削除ができる.
12. プレイリスト毎に音量設定ができる.
13. リピート再生ができる.
14. 未知の記録形式にもソフトウェアの自動追加で対応可能.
15. 他のアプリから曲のファイルを操作可能.

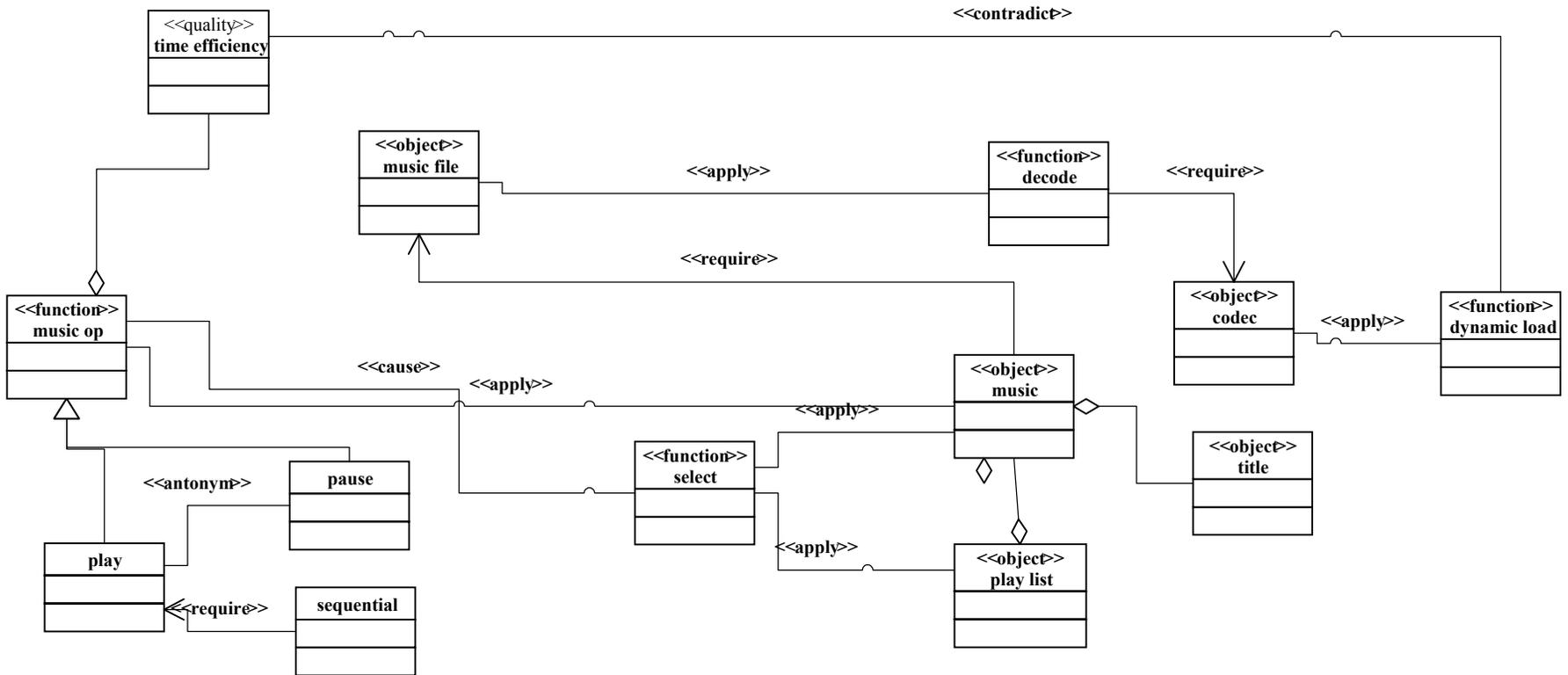
# シソーラス側の構造

- 基本的には型付き有効グラフ
- 枝と節に要求分析に有益な型情報をつけておく.
- シソーラス自体はドメイン専門家ががんばって作る.
  - 本発表では作成法は対象外.

# 型の種類



# シソーラスの例 (部分)



# 推論機構

- 一階述語論理: 既存ツールの利用を鑑みて.
  - 述語: 概念間の関係
  - 変数や定数: 概念
- 論理式
  - 要求文に対応付いたシソーラス要素.
  - 関係固有の推論規則
  - 意味処理したい(証明したい)事項の論理式記述.



# 関係固有の推論規則

- 汎化

$$\forall x \cdot \text{gen}(x, x).$$

$$\text{gen}(x, y) \wedge \text{gen}(y, z) \rightarrow \text{gen}(x, z).$$

- 対語

$$\text{antonym}(x, y) \rightarrow \text{antonym}(y, x).$$

- 他

- $\text{gen}(a, d) \wedge \text{require}(a, c) \rightarrow \text{require}(d, c).$

# 完全性の検出 (1)

- あるobjectに適用可能なfunctionは要求に入れるか否か吟味しましょう.

$$\forall s \forall x \exists y \cdot ((\text{object}(x) \wedge \text{inSpec}(x,s))$$

$$\rightarrow (\text{function}(y) \wedge \text{apply}(y,x) \wedge \text{inSpec}(y,s)))$$

- 上記式をあるオントロジに適用して, 偽ならば, 不完全かもしれないと判定する.

## 完全性の検出 (2)

- すでに要求仕様と言及されている項目に必要なとされている概念は、やはり要求仕様書に記述しましょう。

$$\forall s \forall x \forall y \cdot (\text{InSpec}(x,s) \wedge \text{require}(x,y) \rightarrow \text{InSpec}(y,s))$$

- 同様にこれの真偽を判定する。

# 無矛盾性の判定

- contradictで関係付いている概念同士は同時に仕様書に入らないようにしましょう.

$$\forall s \forall x \cdot (\text{InSpec}(x,s)$$

$$\rightarrow \exists y \cdot (\text{InSpec}(y,s) \wedge \text{contradict}(x,y))$$

- 上記が真なら矛盾があります.

# 意味考慮したメトリクス

- 要求仕様書が意味的にどの程度、完成度があるかの指標となる.
- IEEE830標準の中の意味に関係ありそうな、
  - 妥当性, 完全性, 無矛盾性, 非曖昧性についてオントロジーを使って定義した.

# 定義

$$\text{Correctness} = \frac{|\{x \mid x \in \text{ReqItem} \wedge F_{\text{int}}(x) \neq \phi\}|}{|\text{ReqItem}|}$$

$$\text{Completeness} = \frac{|\{x \mid x \in \text{Con} \cup \text{Rel} \wedge \exists y : \text{ReqItem} \cdot x \in F_{\text{int}}(y)\}|}{|\text{Con} \cup \text{Rel}|}$$

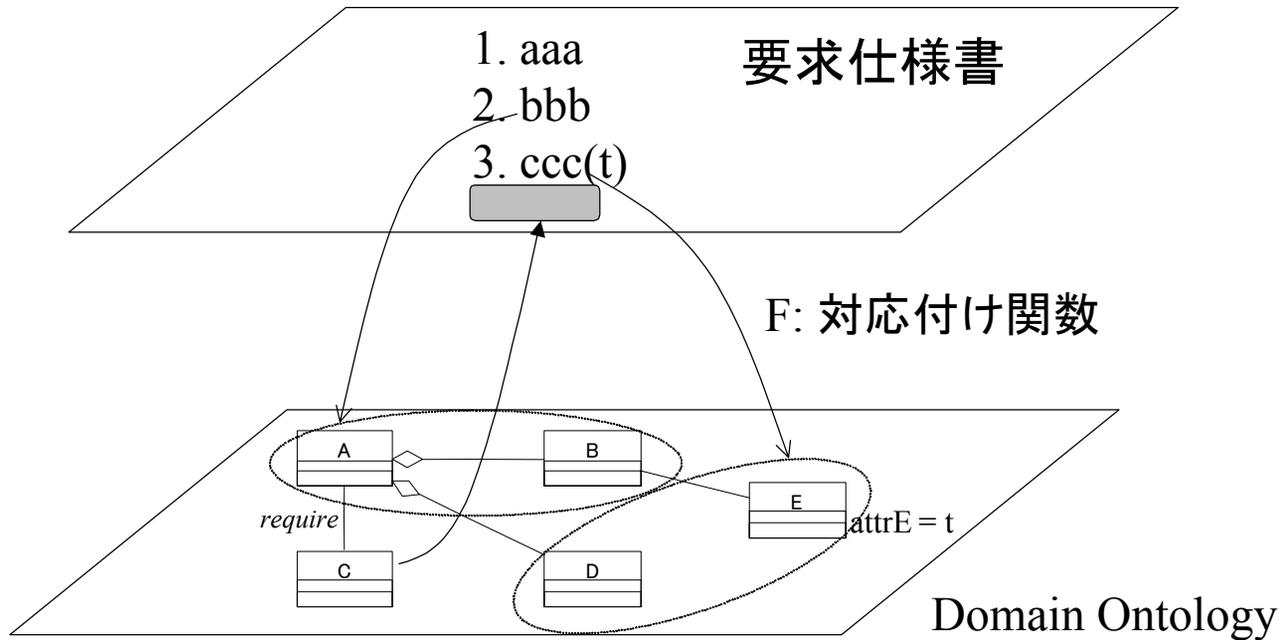
$$\text{Consistency} = \frac{|\{x \mid x \in \text{RCC} \wedge \neg \text{contradict}(x)\}|}{|\text{RCC}|}$$

$$\text{RCC} = \{r \mid \exists a \exists b : \text{Con} \cdot \exists x \exists y : \text{ReqItem} \cdot a \in F_{\text{int}}(x) \wedge b \in F_{\text{int}}(y) \wedge r(a, b)\}$$

$$\text{Unambiguity} = \frac{|\{x \mid x \in \text{ReqItem} \wedge F_{\text{int}}(x) \subseteq \text{Clo}\}|}{|\text{ReqItem}|}$$

Cloはcontradictとantonym以外の推移的閉包

# 要求仕様書とオントロジの関係



説明用に再掲載

# 事例: 以下の要求仕様のメトリクス

1. 音楽の再生できる.
2. 一時停止できる.
3. 前後曲の頭だしができる.
4. 曲操作は早い.
5. 音楽の早送り, 巻き戻しができる.
6. 音量調整と消音ができる.
7. 曲毎に音量設定ができる.
8. 再生速度を変更できる.
9. 再生音程を変更できる.
10. プレイリストのランダム再生ができる.
11. プレイリストに関して削除ができる.
12. プレイリスト毎に音量設定ができる.
13. リピート再生ができる.
14. 未知の記録形式にもソフトウェアの自動追加で対応可能.
15. 他のアプリから曲のファイル进行操作可能.



# 結果

- 妥当性 =  $13/15 = 87\%$ 
  - 8と9がマップできない.
- 完全性 =  $51/(48+67)=44\%$
- 無矛盾性 =  $35/36=97\%$
- 非曖昧性 =  $(15-2)/15=87\%$ 
  - 11と13が複数の無関係な要素にマップできてしまう.

概ね妥当に見えるが問題もある.

# 変更予測

- 頻繁に起こるパターンを収集することで、要求仕様の変更予測を行う。
- メリット
  - 要求項目の安定, 不安定のめぼしがつく。
  - 予測を先取りした要求獲得・定義。

# パターンの収集と評価

- ソフトウェアの変更履歴からパターンを収集.
  1. 機能の新規追加
  2. 既存機能の高機能化
- これらのパターンはオントロジー上の推論規則として記述できそうな目星が付いた.  
(詳細は予稿参照)
- 変更履歴中で追加された機能(12個)のうち, 10個は後に品質の変更が行われた.  
⇒ パターン2は実際に頻繁にありそうだ.

# まとめ

- オントロジを用いて要求仕様書の分析を行う手法の提案.
  - 要求項目をオントロジに写像する.
  - 写像した要素上の単純処理で意味まで考慮した分析を行う.
    - 妥当性, 完全性, 無矛盾性, 非曖昧性や変更予測等
  - 処理は一階述語論理を用いる.
- ソフトウェア音楽プレイヤーを題材としたケーススタディ

# 今後の課題

- オントロジ構築法の確立.
  - PAOREでの成果を流用.
- 支援ツールの開発.
  - グラフエディタ (既存ツール上に構築?)
  - 推論エンジン (Prolog等既存ツールを利用)
- 要求分析以外の工程との連携.
  - 実装技術のオントロジーを構築・連携.