

異なるコンテキストモデリング言語間の 変換手法の提案

早稲田大学

中村 善行

y-naka@nii.ac.jp

開発における問題点

ユビキタス環境では、コンテキスト(現実世界の情報)をモデル化することが重要である。しかし、モデリング言語には特徴があり、環境により適した言語が異なる。そのため、言語毎の記述能力とそのミスマッチを系統的、網羅的に分析し、言語間のデータ変換を行うことが必要となる。

手法・ツールの適用による解決

言語毎の記述能力を分析する際に、分解基準を言語間で統一して分解する。また、言語間のミスマッチを分析する際に、分析の網羅性・完全性を定義し、系統的に分析を行うことで分析の網羅性・完全性を保証する。分析されたミスマッチを分類し、変換パターンを抽出する。

モデリング言語の概要と記述能力分析

➤CSCP

Key-Value形式を階層構造で表現する方式

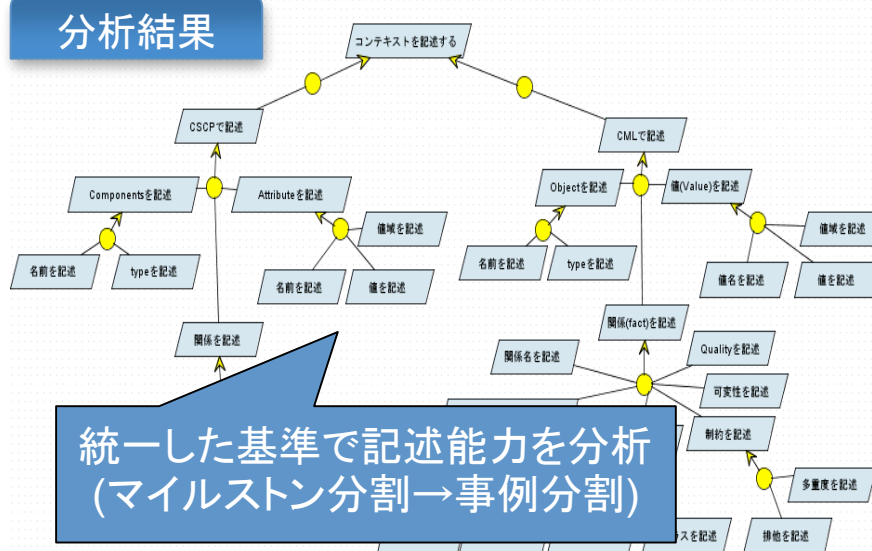
➤CML

オブジェクトロールモデルを用いて表現する方式



相互運用が求められる
(言語間のデータ変換が必要)

分析結果



統一した基準で記述能力を分析
(マイルストーン分割→事例分割)

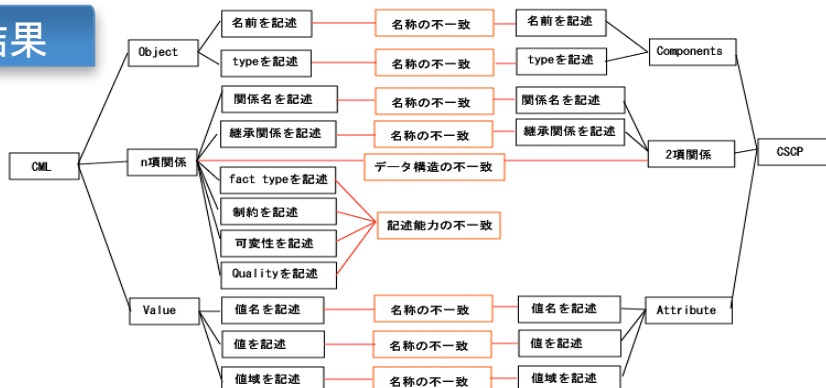
ミスマッチ分析

ミスマッチ分析における完全性・網羅性の定義

全ての表現能力が以下の条件を満たす

- 他方の言語の表現能力と対応
- 他方の言語とのミスマッチとして定義される

分析結果



データ変換パターン抽出

- ミスマッチパターン: 名称の不一致
例: CML: Value, CSCP: Attribute
解決: 名称のマッピングを持たせる
- ミスマッチパターン: 記述能力の不一致
例: 排他性(制約条件)の記述
解決: ミスマッチの機能に相当するメソッドを変換時に織り込む
- ミスマッチパターン: データ構造の不一致
例: 2項関係とn項関係
解決: ・n項関係を複数の2項関係に分解
・n項関係のデータをまとめる