

コンテキストウェアサービスモデルの設計方法の提案と評価

2011SE049 久松 滉平 2011SE064 市川 裕也 2011SE261 棚瀬 泰宏

指導教員 青山 幹雄

1. 研究背景

コンテキストウェアサービスの実現に向けて、環境の変化などの動的なコンテキストデータから、ユーザの意図を予測し、ユーザの意図に適したサービス提供が求められている。本稿では、コンテキストウェアサービスを実現する上で、ユーザの意図に対応したサービスモデルの設計方法と、サービスモデルと機械学習を用いたアーキテクチャを提案する。

2. 研究課題

2.1. 意図に対応したサービスモデルの表現

ユーザを取り巻く動的なコンテキストは、ユーザの意図を変化させる要因になる。しかし、意図の概念の形式的な表現方法は一般化されていない。本稿では、意図の概念の形式的な表現方法を提案する。

2.2. コンテキストデータのフィルタリング方法

処理時間は、コンテキストデータの量に依存することが考えられる。前節のサービスモデルを用いて、意図と関係する必要なコンテキストデータを選別するためのフィルタリング方法を提案する。

3. 関連研究

3.1. 意図に基づくコンテキストウェアサービスのモデル化

コンテキストの変化とドライバの意図の変化からドライバの意図を推測し、適切なサービスを提供するモデルが提案されている[5]。この研究では、コンテキストをベクトル化しコサイン尺度を用いて類似するサービスを提供している。

3.2. オントロジー

ある領域での経験則や知識体系などの概念を明示的にすることである[6]。さらにセマンティック Web の提唱により、その概念を Web 上で情報処理可能な形式で定義し、オントロジーの領域に応じてシステム開発を行うことを可能とした。表現方法として RDF がある。RDF は主語、述語、目的語で表現可能で、リソース間の関係は述語を用いて表現される。

3.3. 機械学習

機械学習を実現する技術の 1 つであるレコメンドを用いた予測分析は、入力データと学習データの類似度を計測することにより、ユーザに適した情報を提供する[2]。実現技術として、オープンソースの Jubatus がある[4]。Jubatus は、Hadoop 連携による並列分散処理によって大量のデータをリアルタイムで処理させることが可能である。

4. アプローチ

運転時のドライバを対象としたコンテキストウェアサービスの実現について、課題に対するアプローチの全体像を示す(図 1)。

4.1. 意図に対応したサービスモデルの表現

意図に対応したサービスのモデル化に関して、対象世界の関係定義と Web 上での情報処理可能なオントロジーに着目し、ドライバ

の意図と関係するコンテキストデータを定義する。それによりドライバの意図に対応したサービスモデルを構築する。

4.2. Jubatus を用いたコンテキストデータの選別

コンテキストウェアサービスのサービス提供の過程で、大量のコンテキストデータの中からサービスに関するコンテキストデータの選別を目的とするサービス提供の仕組みの構築をする。並列分散可能な機械学習の実現技術である Jubatus 用いて、サービスモデルのコンテキストデータと入力データを比較し、サービスマッチングを実現する。さらに、コンテキストデータのフィルタリングにより、処理時間を短縮する。

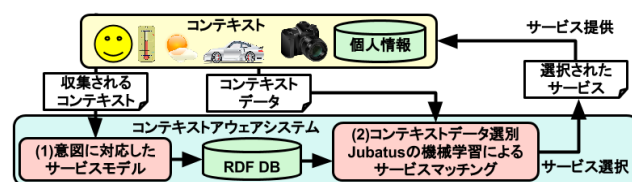


図 1 アプローチの全体像

5. 意図に対応したサービスモデルの表現

5.1. 意図とサービスの関係

ドライバに特化したコンテキストウェアサービスのモデル構築のために、ドライバとサービスの関係を定義した(図 2)。

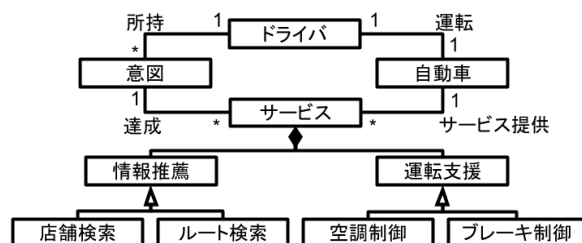


図 2 ドライバの意図とサービスの関係

5.2. 意図の定義

ドライバの意図は動的なコンテキストの変化に依存することから、コンテキストの属性値はドライバの意図を表現できると考えた。また、本稿ではコンテキストウェアサービスのコンテキストの取得方法を理解するために、ドライバの意図を「ドライバに属するコンテキスト」と「周囲を決定付けるコンテキスト」の 2 つに分類した。

5.3. オントロジー構築プロセス

前節までの概念を基に、コンテキストウェアサービスにおけるドライバの意図と関係するコンテキストデータを表現するオントロジーを構築する。以下に、構築プロセスを示す。

(1) 対象の決定

1) 作成するオントロジーの決定

ドライバが対象のコンテキストウェアサービスは多様である。本稿では、オントロジー構築の範囲決定として、カーナビを用いてサービス提供をするコンテキストウェアサービスを対象にする。

2) コンテキストの決定

Dey らのコンテキストの定義[1]を参照して、サービスに適したコンテキストを決定していく。また、本稿ではドライバの状態などを把握できる 5W の概念[3]を基本要素として決定していく。

(2) オントロジーの設計

(1)で決定した定義を元に、オントロジーを構築する。構築には、オントロジーエディタの「法造」を用いる。

(3) オントロジーの更新

設計したオントロジーに対して見直しをする。追加概念が存在する場合は、議論する必要がある。見直した概念に対してオントロジーの更新を行う。

5.4. サービスの RDF モデル

前節で説明したオントロジーを RDF で定義することにより、収集されたコンテキストデータの持つ属性と、コンテキストデータ選別のためのフィルタリングを定義する。

オントロジーでは汎化関係を定義可能なため、本稿ではサービスの汎化関係を RDF スキーマの subClassOf で定義する。サービス毎に関係のあるコンテキストデータは、ユニークな名前空間を用いた述語で定義し、目的語を述語に関するコンテキストデータの属性値として定義する。

6. 提案アーキテクチャ

本稿では、コンテキストアウェアサービスの処理過程で、サービスに関するコンテキストデータを選別するためのフィルタリングブローカを定義する。フィルタリングブローカとは、サービス RDF モデルを用いてドライバとサービス提供者間の仲介をする。

6.1. 提案アーキテクチャの機能

フィルタリングブローカを適用した、提案するフィルタリングアーキテクチャの機能を示す(図 3)。

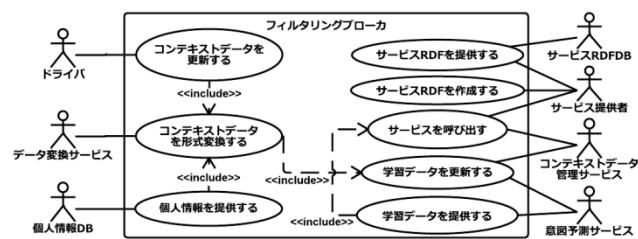


図 3 フィルタリングアーキテクチャの機能

フィルタリングブローカを構成するサービスを以下に示す。

- (1) 意図予測サービス: Jubatus のレコメンドを用いてサービス RDF モデルから取得した学習データを基に、入力されるコンテキストデータからドライバの意図に対応したサービスを選択し、処理結果をサービス提供者者に送信する。
- (2) コンテキストデータ管理サービス: サービス RDF モデルが更新された際に、学習データを更新するサービスである。
- (3) データ変換サービス: コンテキストデータを処理可能な形式に変換するサービスである。主に RDF の操作を行う。

6.2. コンテキストアウェアサービスのフィルタリングアーキテクチャ

フィルタリングブローカを適用したフィルタリングアーキテクチャの構造を図 4 に示す。

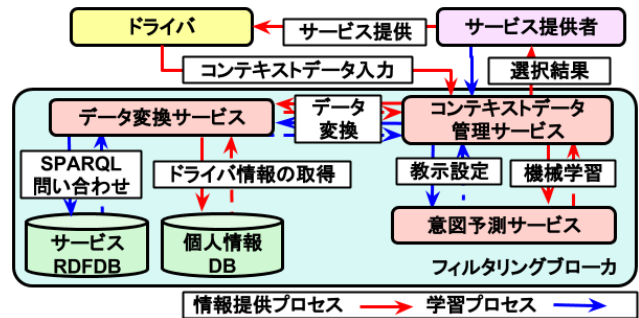


図 4 フィルタリングアーキテクチャの構造

6.3. アーキテクチャの振る舞い

- (1) 学習プロセス: サービス提供者がサービス RDF を更新した際に、教示学習の変更を行う学習データ更新プロセスである(図 5)。

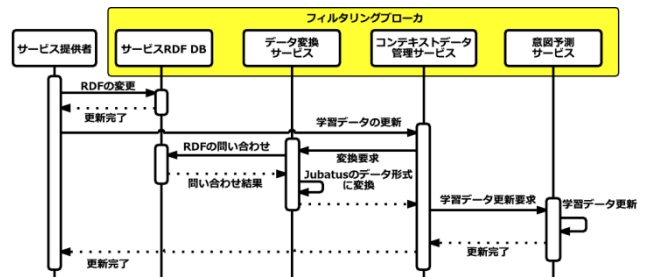


図 5 学習プロセスの振る舞い

- (2) サービス提供プロセス: アーキテクチャにおけるシステムの振る舞いを図 6 に示す。前提条件として、レコメンドに必要な学習データは更新しているものとする。

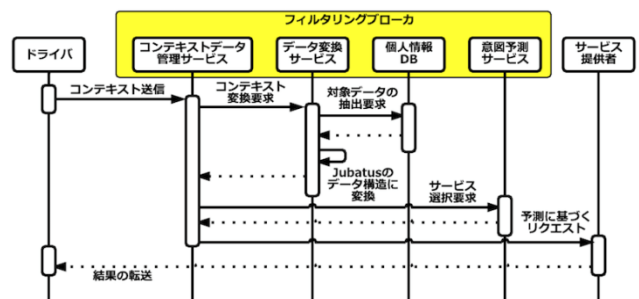


図 6 提供プロセスの振る舞い

7. プロトタイプ

7.1. プロトタイプの目的

本稿がプロトタイプを開発する目的を以下の 3 つである。

- (1) サービス RDF モデルを用いたコンテキストデータ選別によるリアルタイム性の確認
- (2) コンテキストアウェアサービス実現の妥当性の確認
- (3) 意図に対応したサービスモデルの表現の有用性の確認

7.2. プロトタイプの要求

運転時のコンテキストデータの変化に対応したサービスを提供する提案アーキテクチャの適用による、データ処理のリアルタイム性の向上と、モデルの評価を行う。

7.3. プロトタイプ構成と環境

プロトタイプの構成を図7に示す。

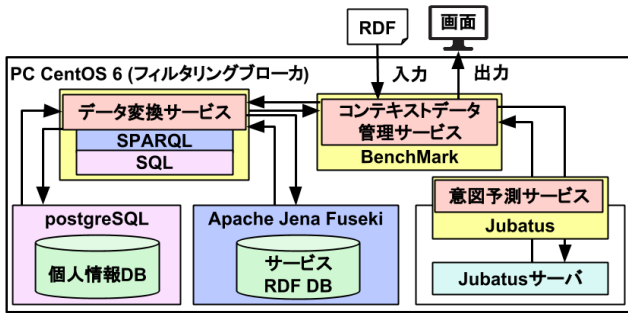


図7 プロトタイプの構成

ブローカシステムのRDBにpostgreSQL, RDFDBにFusekiを使用した。また、Jubatusで転置インデクスアルゴリズムを使用することにより、式(1)を用いて類似度を算出する。

$$\cos(x, y) = \frac{x^T y}{|x||y|} \quad (1)$$

表1 プロトタイプの実行環境

OS	CentOS release 6.5 (Final)
メモリ	1GB
CPU	Intel(R) Core(TM)2 Duo CPU 2.66GHz
Jubatus	jubatus-0.5.4 (jubarecommender)
PostgreSQL	psql (PostgreSQL) 8.4.20
Apache Jena Fuseki	Fuseki 1.1.1

7.4. オントロジーモデルの設計

上位概念の「時間, 場所, 物質, 実在物(抽象物, 具体物, プロセス), 属性, 量」を基に汎化していく。設計した全体のオントロジーを図8に示す。

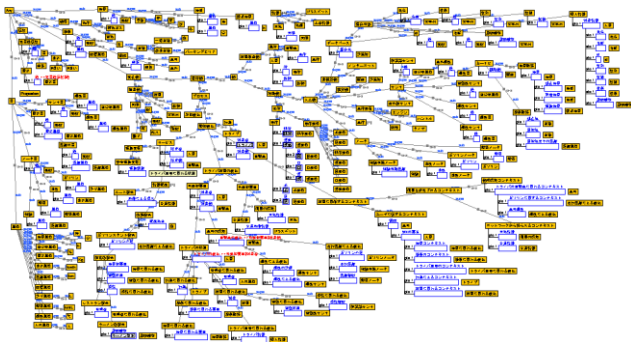


図8 ドライバの意図と関係するコンテキストデータを表現するオントロジー

7.5. サービスRDFモデルの設計

サービス提供に関するコンテキストデータを用いたサービスRDFモデルを定義する(図9)。本稿でのサービスは、「コンビニ検索」、「ガソリンスタンド検索」、「迂回ルート検索」、「レストラン検索」とする。

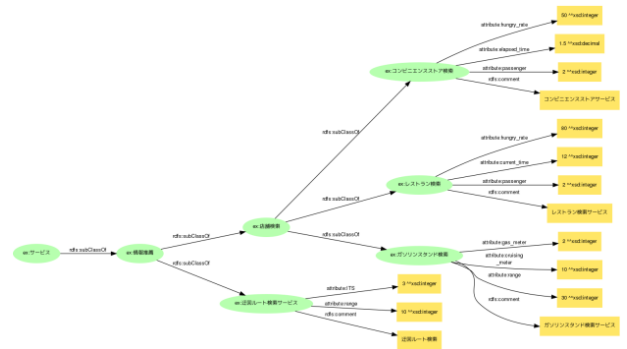


図9 サービスRDFモデル

7.6. プロトタイプの振る舞い

プロトタイプにおける意図予測サービスまでのプロセスの振る舞いを図10で示す。前提条件として、サービスRDFモデルは更新してあるものとする。処理時間の計測は、コンテキストデータを意図予測サービスに入力してからサービスを選択するまでとする。

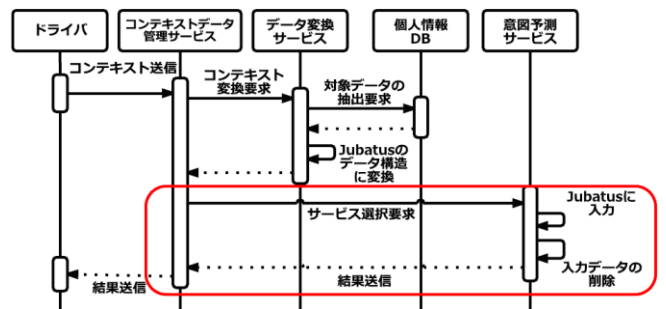


図10 プロトタイプにおけるサービス提供プロセスの振る舞い

8. 例題への適用

8.1. 例題の目的

本稿では、ドライバへのサービス提供の実現と、サービスモデルの改善のために2つのシナリオを用いる。

シナリオ1では、例題を適用することによるコンテキストウェアサービス実現の妥当性とリアルタイム性の検証する。

シナリオ2では、ドライバは、プロトタイプによる「レストラン検索サービス」を利用せず行きつけの「ラーメン店」に向かった。このことから、オントロジーの更新によるオントロジー構築プロセスの有用性を検証する。

8.2. シナリオ1: サービス提供の実現

- (1) 1月20日, 気温 10°C, 季節冬, ドライバが南山大学瀬戸キャンパスを出て, 名古屋駅に向けて正午 12 時に出発する。
- (2) 20 分後, 愛地球博公園周辺で空腹になる。
- (3) さらに 40 分後, 本郷駅でガソリンが 2L になる。

8.2.1. 適用結果

「南山大学」から「名古屋駅」までの経路とサービス提供の実行結果を示す(図11)。



図 11 適用結果

8.3. シナリオ 2: サービスモデルの改善

8.3.1. 適用結果

プロトタイプは、収集されたコンテキストデータより「レストラン検索サービス」を提供したが、レストラン検索サービスを利用せず行きつけのラーメン店に向かった。

8.3.2. オントロジーモデルの更新

シナリオ 2 の結果から、プロトタイプはドライバに「レストラン検索サービス」を提供したがドライバはラーメン店に向かい、ドライバに最適なサービス提供されなかった。

オントロジー構築プロセスはシナリオ 2 の結果より、オントロジーの個人情報に「趣味嗜好」という基本概念を追加する。そして、情報推薦に「ラーメン店検索」を追加する(図 12)。

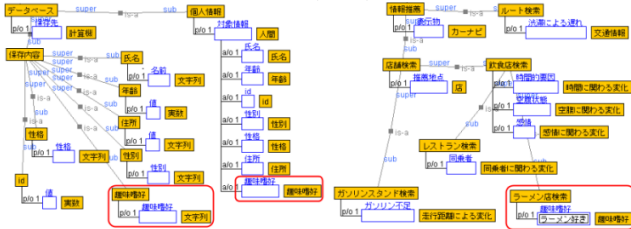


図 12 更新されたオントロジー

8.4. 意図予測サービスの処理時間の計測

本稿では、シナリオ 1 を使用して処理時間の計測を行う。一般的な n 行 k 列の類似度の計算量は $O(kn^2)$ となるため、比較対象の多さにより処理時間に影響を与える。Jubatus では、Key-Value 形式のデータ構造をしており、Key を指定することにより処理する対象を定義することが可能である。処理時間の計測方法として Ruby の Benchmark ライブラリを使用した。

表 2 計測結果

フィルタリング	回数	平均値(ms)	標準偏差	削減率	類似度
あり	10	1.14	0.008	0.91	0.93
なし	10	1.24	0.010	1.00	0.35

9. 評価と考察

9.1. 意図に対応したサービスモデルの表現の評価

ドライバの意図に対応したサービスのオントロジーを構築することで、サービス提供に必要な知識の深さと広がり大きいことが明らかになり、サービス提供に必要な知識モデルを表現することができた。そして、プロトタイプの実装により、オントロジー全体の基本概念数が 2 つ、ロール概念数が 3 つ増加した(表 3)。よって、オントロジー全体の基本概念数、ロール概念数の増加によるオントロジー構築プロセスの有用性を確認した。

表 3 例題の適用による概念数の比較

	シナリオ 1	シナリオ 2	増加数
基本概念の個数	171	173	2
ロール概念の個数	120	123	3

9.2. Jubatus を用いた提案アーキテクチャの評価

(1) Jubatus のレコメンドと類似度を用いたサービス提供の評価

サービスに関するコンテキストデータを、RDF を用いて定義した。それを用いて Jubatus のレコメンドに適用した機械学習によりサービス提供を行なった。これにより、提案アーキテクチャのコンテキストウェアサービス実現の妥当性を確認した。

(2) Jubatus を適用することによるリアルタイム性の評価

リアルタイム性を向上させるために、処理に必要なある Key の部分を Jubatus のデータ構造で記述することによりコンテキストデータのフィルタリングを実現した。

例題への適用では、処理時間を計測するために Ruby の Benchmark ライブラリを使用して計測し、フィルタリングありとなしでは、処理時間を 9.0% 削減した。さらに、機械学習によるコンテキストデータ選別により、類似度が 1 に近づいた(表 2)。類似度が 1 に近づくほどドライバに最適なサービスが提供されるため、サービス提供の正確性が向上した。

10. 今後の課題

提案アーキテクチャを適用したプロトタイプを用いて、運転時のサービス提供によるコンテキストウェアサービスを実現した。しかし、提案方法の汎用性向上のため、多様なサービスについても定義する必要がある。本稿では、意図予測サービスは類似度の差によってサービス選択されるため、シナリオによってはドライバに対して最適なサービスが提供されない可能性がある。よって、サービス選択されるための条件を追記することでサービス提供の精度が向上すると考えられる。

11. まとめ

概念を定義するオントロジーに着目しドライバの意図に対応したサービスモデルを表現した。そして、提案アーキテクチャの適用による機械学習とサービス RDF モデルを用いたコンテキストデータ選別のためのフィルタリングをした。提案アーキテクチャを利用したプロトタイプを作成し、例題を用いてサービス提供の処理時間を計測した。さらに、サービス提供に反映されていない概念を追記することにより、オントロジー構築プロセスの有用性を検証した。

参考文献

- [1] D. Allemang, Semantic Web for the Working Ontologist, Morgan Kaufmann, 2008.
- [2] D. Conway, et al, Machine Learning for Hackers O'Reilly, 2005.
- [3] 藤波 香織, 分散コンテキストウェアシステムに関する研究, 早稲田大学院理工学研究科博士論文, 2005.
- [4] Jubatus, オンライン機械学習向け分散処理フレームワーク, <http://jubatus/ja/>, accessed(2015/1/1)
- [5] 牧 慶子ほか, 意図に応じたコンテキストウェアサービス提供モデルの提案と評価, 情報処理学会, 2012.
- [6] 溝口 理一郎, オントロジー構築入門, オーム社, 2006.