

SMD に対する Linked Data に基づく リソース指向アーキテクチャ設計法の提案

2008MI160 中根 拓也 2009SE026 福田 雄介 2009SE093 岩下 翔

指導教員: 青山 幹雄

1 はじめに

SMD(Smart Mobile Device)[2]の普及に伴い, SMD に対応した Web サービス開発の基盤として REST の利用が広がっている. しかし, REST(REpresentational State Transfer)[3]に基づくアーキテクチャ設計法が明確に定義されていない問題がある. また SMD のメモリや CPU の処理能力の制約が考慮されていない問題がある.

本研究では, SMD の環境に適した Web サービス開発のため, REST に基づく, RDF(Resource Description Framework)[5]を用いたアーキテクチャ設計法を提案する.

2 研究課題

本研究では, Web サービスを SMD で利用する場合の SMD の制約された環境という問題を解決するために, REST に基づくアーキテクチャの設計法を提案する.

SMD の課題である制約された環境に対応するために, 送受信データの容量を抑える必要がある. そこで RDF のデータモデルに着目し, クライアントの要求に関連性のあるデータを抽出し, 送信する. さらに, SMD に対応した Web サービス開発を実現するために REST に基づくアーキテクチャの設計法を提案する.

3 関連技術

3.1 SMD

SMD とは, 従来のモバイルデバイスより演算能力, 処理能力が優れているデバイスで, Web に常時接続可能なものを指す. Wi-Fi や 3G 回線などの接続方法があるため, どこでもネットワーク接続が可能である. しかし, メモリや CPU の処理能力に制約があり, かつネットワークの接続が不安定である.

3.2 リソース指向アーキテクチャ

リソース指向アーキテクチャ(ROA: Resource-Oriented Architecture)とは, REST の設計原則に基づいており, Web 技術を使ったアーキテクチャである. REST とは, HTTP を用いた対話モデルを定義するアーキテクチャスタイルである. 4 つの設計原則として HTTP を使用, ステートレス, URI で参照可能, XML 等で表現がある[1].

3.3 Linked Data

3.3.1 RDF

RDF とは, リソースのメタデータの記述言語である. リソースを主語(Subject), 述語(Predicate), 目的語(Object)のトリプル(三要素)で表現する. RDF はマシン可読のスキーマを使っているため, RDF モデルを用いることで, 情報の分類や検索などの自動化, 効率化が可能である.

3.3.2 SPARQL

SPARQL(SPARQL Protocol And RDF Query Language)[6]とは, RDF 探索に用いるクエリ言語である. rdfDB, RDQL, SeRQL 等の従来の RDF クエリ言語の上に構築されており, さらに SPARQL 独自の機能を含んでいる. SPARQL には SELECT, CONSTRUCT, ASK, DESCRIBE の 4 つのクエリ形式がある.

4 アプローチ

本研究では, SMD の制約された環境に対応するため, 以下のアプローチをとる.

4.1 REST に基づくアーキテクチャ設計法

SMD に対応した Web サービス開発を実現するため, REST の単純な構造に着目し, REST に基づくアーキテクチャ設計法を提案する. REST には SMD の環境に適した以下の特徴がある.

(1) URI を使用

実データの送受信より, データ量が削減可能である.

(2) ステートレス

ネットワークの不安定さの影響を抑える.

4.2 RDF グラフを用いたデータ抽出

RDF モデルの情報の探索, 抽出の容易さに着目する. RDF 形式で保存されているデータから SPARQL を用いて情報を探索, 抽出する. これにより送受信するデータ量を抑制できる.

5 提案アーキテクチャ

5.1 提案アーキテクチャの構成

REST を基盤とし, RDF を用いたアーキテクチャを提案する(図 1).

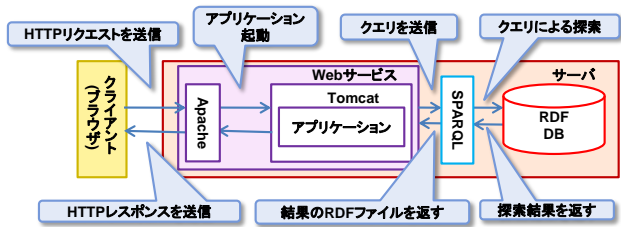


図1 提案アーキテクチャ

このサービスではデータは RDFDB に保存されている。クライアントからは SPARQL クエリが送信される。

5.2 アーキテクチャの振る舞い

提案アーキテクチャの振る舞いを図2に示す。

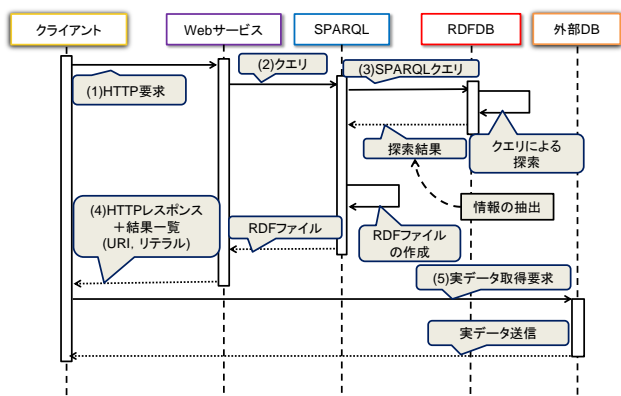


図2 アーキテクチャの振る舞い

- (1) クライアントは HTTP 通信でリクエスト URI とクエリを Web サービスに送信する。クエリには取得したいリソースと、どのような関連性のあるリソースを取得したいかを、RDF のプロパティで記す。
- (2) Web サービスはクライアントから送信されたクエリを SPARQL に送信する。
- (3) SPARQL クエリを用いて RDFDB 内の RDF グラフを探索する。SPARQL は探索結果の複数のトリプルを1つの RDF グラフに結合することで RDF ファイルを作成する。この作成した RDF ファイルを Web サービスに送信する。
- (4) Web サービスは HTTP 通信でクライアントに URI とリテラルを含んだ結果一覧を送信する。
- (5) クライアントは Web サービスから送られてきた結果を基に、実データが保存されている外部 DB に実データ取得の要求を送り、データを取得する。

5.3 情報の探索と抽出

送信データ量抑制のために、SPARQL を用いた RDFDB の情報探索と抽出を行う。RDF モデルはトリプルを基本構造とした RDF 文を複数組み合わせ合わせたグラフ構

造であり、要素間の関係性が URI で繋がっている(図3)。

SPARQL を用いて要素間の関係性をたどることで関連性のあるデータを特定し、抽出する。

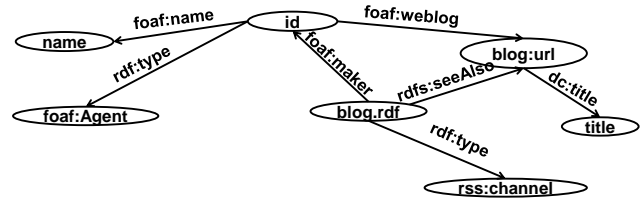


図3 RDF モデルの構造

6 プロトタイプ

SPARQL を用いた RDF ファイルのデータ探索と抽出の妥当性確認のため、プロトタイプを実装した。アプリケーションサーバとして Tomcat, RDF データ管理に Sesame[4]を用いる(図4)。

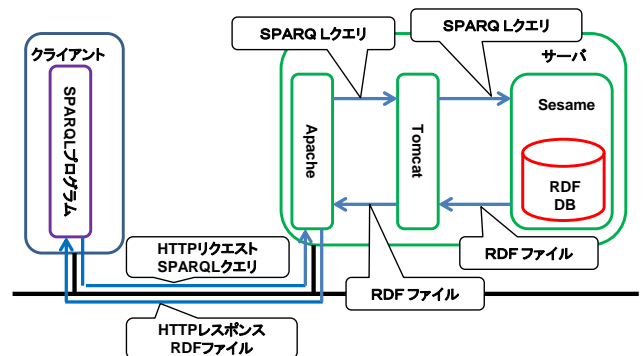


図4 システム構成

表1 開発環境

	クライアント	サーバ
OS	Windows Vista SP2	Windows Vista SP2
メモリ	4.00GB	2.50GB
CPU	Intel(R) Core(TM)2 Duo 2.00GHz	Intel(R) Pentium(R) Dual CPU 1.60GHz

表2 ソフトウェア環境

	クライアント	サーバ
Apache		Apache 2.2.23
Tomcat		Tomcat 6.0.36
Sesame		Sesame 2.2.3
SPARQL	2008/01/15 勧告	2008/01/15 勧告

7 プロトタイプによる評価

RDF の構造の特徴から、探索階層の増加、取得リテラル数の増加により応答時間の増加が考えられる。その応答時間の増加傾向から評価を行う。

7.1 探索階層による応答時間の変化

1 個のリテラルを 1, 5, 10, 20 階層から取得する。それぞれ 100 回実行し、応答時間を測定した。ガベージコレクションによる特異値(3σ 以上)を除いた平均応答時間を補正值とした。その結果を図 5 に示す。

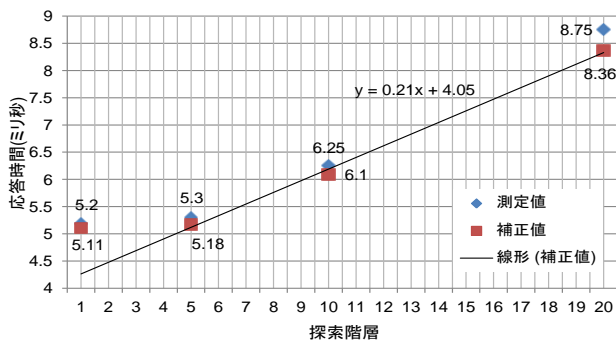


図 5 探索階層による応答時間

表 3 平均応答時間と標準偏差

探索階層	1	5	10	20
平均応答時間(ミリ秒)	5.11	5.18	6.10	8.36
標準偏差	0.78	0.96	0.98	1.65

7.2 取得リテラル数による応答時間の変化

1 階層から 1, 5, 10, 20 個のリテラルを取得する。それぞれ 100 回実行し、応答時間を測定した。ガベージコレクションによる特異値を除いた平均応答時間を補正值とした。その結果を図 6 に示す。

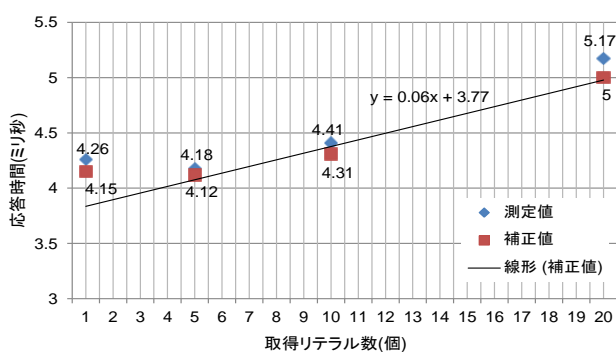


図 6 取得リテラル数による応答時間

表 4 平均応答時間と標準偏差

取得リテラル数	1	5	10	20
平均応答時間(ミリ秒)	4.15	4.12	4.31	5.00
標準偏差	0.85	0.69	0.88	0.88

7.3 探索階層, リテラル数による応答時間の変化

取得リテラル数 1, 5, 10, 20, 100 個それぞれを 1, 5, 10, 20 階層から取得する。それぞれ 100 回実行し、応答時間を測定した。ガベージコレクションによる特異値を除

いた平均応答時間を補正值とし、補正值の線形近似をとった。その結果を図 7 に、線形近似直線の傾きと切片を表 7 に示す。

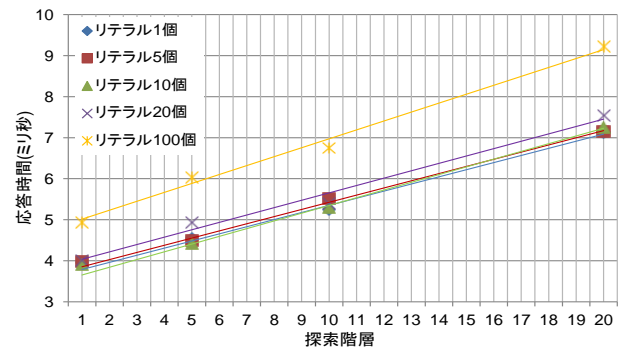


図 7 リテラル数, 探索階層による応答時間

表 5 平均応答時間

	1 個	5 個	10 個	20 個	100 個
1 階層	4.05	4.04	4.01	4.14	5.11
5 階層	4.65	4.61	4.55	5.16	6.24
10 階層	5.46	5.78	5.56	5.68	7.06
20 階層	7.57	7.58	7.68	8.06	9.63

表 6 標準偏差

	1 個	5 個	10 個	20 個	100 個
1 階層	0.81	0.60	0.78	0.86	1.13
5 階層	0.93	0.98	1.02	1.15	1.10
10 階層	1.32	1.48	1.32	1.56	1.51
20 階層	1.95	1.94	1.88	1.98	2.08

表 7 リテラル数ごとの線形近似直線の傾きと切片

	1	5	10	20	100
傾き	0.17	0.17	0.19	0.18	0.22
切片	3.62	3.68	3.47	3.68	4.80

7.4 評価

応答時間の計測により以下のことが示された。

- (1) 探索階層の増加により応答時間が線形的に増加する。探索階層が 1 階層増加すると約 0.2 ミリ秒応答時間が増加する。
- (2) 取得リテラル数の増加により応答時間が線形的に増加する。取得リテラル数が 1 個増加すると約 0.06 ミリ秒応答時間が増加する。
- (3) 取得リテラル数, 探索階層, 両方の増加により応答時間が非線形的に増加する。リテラル 1 個の傾きは 0.17 であるが, リテラル 100 個の傾きは 0.22 であり, 傾きが 0.05 増加している。

応答時間の増加が探索階層 1 階層増加では約 0.2 ミリ

秒に対し、取得リテラル数 1 個増加では約 0.06 ミリ秒である。そのため探索階層増加による応答時間の増加は、取得リテラル数増加による応答時間の増加に対し大きい。

RDFDB の探索はサーバで行っているため、SMD から Web サービスを利用する環境であっても探索階層による応答時間の変化は実験結果と同様であると思われる。また SMD から Web サービスを利用する環境では、ネットワークが不安定である。そのため、取得リテラル数による応答時間の変化は実験結果より少し増加すると思われる。しかし、取得リテラル数増加による応答時間の増加は小さいため、増加量は少なく、応答時間に対する影響は少ないといえる。

また、取得リテラル数が増加すると探索階層による応答時間の増加量が大きくなる。しかし、(3)から応答時間の増加量は小さいことが分かる。

8 考察

8.1 REST に基づくアーキテクチャ設計法

現状の REST に基づく Web サービスは、アーキテクチャが明確に定義されていない。そのため、開発者ごとに同一のサービスであっても、アーキテクチャが曖昧なため体系的な設計を行えない。REST に基づくアーキテクチャ設計法の提案により、体系的な Web サービスの開発が可能となる。

8.2 RDF データの探索と抽出

(1) SPARQL を用いた RDFDB の探索

SPARQL を用いて RDFDB へ探索と抽出を行う。探索と抽出によりユーザの必要な情報が抽出される。それにより、ユーザに送信するデータ量が抑制可能である。

(2) URI の取得

RDF はリソースの URI を記述する。そのため、ユーザは探索結果を URI で取得する。これにより、一度に受信するデータ量の抑制が可能である。

(1)、(2)から大量データの送受信にかかる時間が抑制される。そのため、SMD の制約された環境に対し有効である。しかし、データリンク数の増加により探索時間が増加する。そのため探索時間を短縮するように RDF のデータ構造を作成する必要がある。

8.3 RDF データアクセスの性能評価

RDFDB 探索の探索階層、取得リテラル数の変化による応答時間を測定し、評価を行った。探索階層の増加、取得リテラル数の増加、共に応答時間が増加したが、増加量は少ない。100 個のリテラルを 20 階層からの取得にかかる応答時間が 9.63 ミリ秒であるので応答時間は十分短いといえる。よって提案するアーキテクチャは妥当性があるといえる。

9 今後の課題

9.1 複数 RDFDB への対応

実装したプロトタイプでは 1 つの RDFDB に対する探索と抽出の応答時間を測定した。しかし、実際の Web サービスでは、複数の DB に RDF ファイルが保存されていることが想定される。よって単一の RDFDB ではなく、複数の RDFDB に対するアーキテクチャ設計法の考察と探索と抽出の応答時間を測定することが必要である。

9.2 Linked Data でない場合

実装したプロトタイプでは探索する DB は RDFDB であることを前提としている。しかし、現状の Web サービスでは RDFDB が用いられていないことが多い。そのため、RDFDB 以外の DB に対応したアーキテクチャ設計法の考察が必要である。

10 まとめ

本稿では、SMD の制約された環境に対応した Web サービス開発を実現するために、SMD に対応するリソース指向アーキテクチャ設計法を提案した。REST の単純な構造、RDF モデルの探索と抽出の容易さに着目し、SMD の制約に対応するためのアプローチを示した。そして、SPARQL と RDFDB を適用した SMD 対応アーキテクチャを提案した。

提案したアーキテクチャの妥当性を確認するためにプロトタイプの実装を行った。プロトタイプを用いて RDFDB の探索と抽出にかかる応答時間の測定、評価を行った。それにより、取得リテラル数、探索階層の増加による応答時間に対する影響は少ないことを示し、提案アーキテクチャの妥当性を示した。

参考文献

- [1] A. Rodriguez, RESTful Web サービスの基本, 2008 <http://www.ibm.com/developerworks/jp/webservices/library/ws-restful/>.
- [2] J. H. Christensen, Using RESTful Web-Services and Cloud Computing to Create Next Generation Mobile Application, Proc. of OOPSLA '09, 2009, pp. 627-634.
- [3] L. Richardson, S. Ruby, RESTful Web サービス, 第 2 版, O'Reilly, 2008.
- [4] T. Segaran, C. Evans, J. Taylor, セマンティック Web プログラミング, 第 1 版, O'Reilly, 2010.
- [5] W3C, RDF 入門, 2005, <http://www.asahi-net.or.jp/~ax2s-kmtn/internet/rdf/rdf-primer.html>.
- [6] W3C, RDF 用クエリ言語 SPARQL, 2008, <http://www.asahi-net.or.jp/~ax2s-kmtn/internet/rdf/rdf-sparql-query.html>.