

セマンティック Web サービスとオブジェクト指向メソッドディスパッチ

Semantic Web Services vs. Object-Oriented Method Dispatch

小出誠二
Seiji Koide

株式会社ギャラクシーエクスプレス
Galaxy Express Corporation

Web Service decomposition and interpretation is similar to the method dispatch mechanism of Common Lisp Object System. In this paper, we compare the both and discuss the equivalency and the difference. Then, we propose an idea of new funcallable object for Web Services. It is implemented using Meta-Object Protocol so as to realize Web Service decomposition and interpretation internally. Programmers will obtain benefits that Web Services can be programmed in lisp code as well as methods and functions. We call this new Web Service programming paradigm Unified Services.

1. はじめに

セマンティック Web サービスにおけるサービス分解・実行では複合 Web サービスをその記述にしたがって解釈し、その中に含まれるアトミックな Web サービスを実行する。OWL-S プロセス記述では If ~ Then ~ Else や Repeat ~ Until のような制御構造があり、サービス分解・実行処理系は記述された制御構造に従ってこれを解釈しながら、同時に抽象サービスから特殊サービスへサービスの IS-A 階層構造を降下しつつアトミックな Web サービスを同定・実行しなければならない。また、当然のことながらそこでは Web サービス呼出しと通常の計算手続きが混在している。以上は OWL-S 仕様に含まれる思想であるが、実際の OWL-S 記述は実用的見地からは低レベル過ぎ、さらに実用的なサービス分解・実行の機構と言語が求められている。

セマンティック Web サービスにおいて抽象複合サービス記述としてビジネスプレートあるいはタスクフローがあらかじめ与えられており、エージェントがそのタスクフローを参照しつつ環境から目的にかなう Web サービスを発見して実行することが考えられる。研究者によってはこれをサービス合成と言うが、ここではビジネスフローやタスクフローのような抽象プレートがあらかじめ与えられている問題は、サービス分解・実行と捉えることにする。本報告では抽象複合サービスの分解・実行問題を、オブジェクト指向プログラミングにおけるメソッドディスパッチとの対比で議論し、メソッドディスパッチの機構ではなく Web サービスの分解・実行を内部に抱えたサービスオブジェクトの実装を提案する。

2. CLOS におけるメソッドディスパッチ

2.1 オブジェクト指向プログラミングと CLOS メソッド

オブジェクト指向システムとしてここでは Common Lisp Object System (CLOS) を取り上げる。CLOS におけるメソッドは Java や C# のようなオブジェクト指向言語と異なり、オブジェクトに付着する手続きではない。それは通常の関数の拡張であり、オブジェクトとは独立に存在する。複数ある同名のメソッド群はまとめて汎化関数 (generic function) と呼ばれる

連絡先: 小出誠二, 株式会社ギャラクシーエクスプレス,
東京都港区浜松町 1 - 18 - 16, TEL:03-5733-7120,
koide@galaxy-express.co.jp

が、入力引数の型の組み合わせによって実際に実行される手続きが決定される (引数の数は同一でなければならない)。

航空券予約やホテルの予約を包含する抽象的な予約手続きを、抽象メソッドとして CLOS で記述すれば次のようになる。

```
(defmethod book ((description Description)
                (cardID CreditCardNumber)
                (uid UserID)
                (session SessionKey))
  ;; This method returns an instance of Booking Code
  (error "An abstract method:book is called."))
```

ここで Description や CreditCardNumber は定義域 (domain) に記述されている概念 (terminology) あるいは OWL クラスである。CLOS 上に開発されたセマンティック Web プロセッサ SWCLOS[Koide] では OWL クラスはそのまま CLOS のクラスに写像されるため、このような CLOS メソッド記述が可能になる。

実際の予約手続きは、第 1 引数に対して航空券に関しては FlightDescription を、ホテルに対しては RoomDescription をとるようなメソッドを記述する。

```
(defmethod book ((description FlightDescription)
                (cardID CreditCardNumber)
                (uid UserID)
                (session SessionKey))
  ;; This method returns an instance of FlightBooking Code
  ;; The followings are method body.
  ...
)
```

```
(defmethod book ((description RoomDescription)
                (cardID CreditCardNumber)
                (uid UserID)
                (session SessionKey))
  ;; This method returns an instance of RoomBooking Code
  ;; The followings are method body.
  ...
)
```

ここで FlightDescription と RoomDescription は Description の下位概念あるいはサブクラスであることを注意しておく。こうすることによりプログラマーは与える第 1 引数の種類を気にすることなく、メソッド book を呼べば CLOS のメソッドディスパッチによってそれが FlightDescription のインスタンスであれば航空券用のメソッドが、RoomDescription のインスタンスであればホテル用のメソッドが起動実行される。

2.2 CLOS メソッドとサービス分解・実行

上記例では、第1引数のみが特殊化されていたが、CLOS のメソッドディスパッチでは第1引数のみならず、すべての引数の型の組み合わせによって実行されるべきメソッドが決定される。この機能は階層的計画手法を用いた Web サービスの分解・実行の働きとよく似ている。SHOP2(Simple Hierarchical Ordered Planner)[Nau] 風の階層的計画システムを用いた場合には、エージェントは指定されたトップの複合サービス(ここでは book に相当)からサービスの IS-A 階層を降下しながら最も特殊なサービスで手持ちの入出力引数に適応可能なサービスを探し出す [小出 2005b]。

しかし、CLOS メソッドディスパッチと階層的計画手法による Web サービス分解・実行には以下のような違いもある。

- CLOS メソッドは大域的あるいは局所的な名前を持ち、同じ名前のメソッド間のみでメソッドディスパッチが考慮されるが、Web サービス分解・実行では名前は決定要素ではなく、指定されたトップの抽象サービスとそのサブクラスサービスが考慮される。
- メソッド引数の個数は対象となるメソッド間で同一でなければならないが、Web サービス分解・実行ではかならずしもそうではない。
- Web サービスでは入出力引数以外に(サービス実行の)前提条件や(サービス実行により生じる副作用である)結果も考慮して Web サービスを特定する必要がある。

3. 関数呼出し可能なオブジェクトによるサービス実装

CLOS の Meta-Object Protocol では汎化関数 (generic function) は `funcallable-standard-class` というメタクラスのインスタンス、`funcallable-standard-object` のサブクラスとなっている。このようなオブジェクトは `funcallable-instance` と呼ばれ、通常関数と同じように `apply` や `funcall` の引数となることができる。CLOS では同名のメソッドはその汎化関数オブジェクトのメンバーになっており、メソッドディスパッチとして実効メソッドが計算され、そのメソッド関数部が実行される。そこで Web サービスも CLOS メソッドと同様に、`funcallable-standard-class` のインスタンス、`funcallable-standard-object` のサブクラスとして定義する。想定されるサービス定義例を以下に示す。

```
(defService Book (Service)
  (inputs (description Description)
          (cardID CreditCardNumber)
          (uid UserID)
          (session SessionKey))
  (outputs (code BookingCode))
  (precondition BookingPrecondition)
  (result BookingResult))

(defService FlightBook (Book)
  (inputs (description FlightDescription)
          (cardID CreditCardNumber)
          (uid UserID)
          (session SessionKey))
  (outputs (code FlightBookingCode))
  (precondition FlightBookingPrecondition)
  (result FlightBookingResult))
```

```
... ;; hereafter the body of service
)
```

もしサービス定義中に通常関数ボディのような記述があれば、引数ラムダリストでラップしてそれをリスプロコードとしてサービスオブジェクトの関数部にセットする。もし関数ボディ記述がなければそれは抽象サービス記述であり、このサービスから IS-A 階層で1段下のサービス群から適応可能なサービスを選択する手続きをサービスオブジェクトの関数部にセットする。リスプロ処理系は `funcallable-object` に対して、そこにセットされている関数部を取り出して実行する。すなわち、クロージャであればそれを実行し、名前つき関数であれば関数定義を取り出して実行し、汎化関数であれば実効メソッドを計算してメソッド定義にある手続きを実行する。もしそれが Web サービスの階層的選択手続きであればそれが実行され、さらに同様な計算が進行される。

こうすることでリスプロコードの見かけ上、Web サービスをあたかも関数と同じように `apply` や `funcall` に用いることができるようになり、XML による煩雑な OWL-S の定義やサービスの解釈実行をプログラマーの目から消し去って、効率的なセマンティック Web サービスプログラミングが可能となる。

抽象サービスと同様にボディ部がないサービスで下位のサービス記述がないサービスはアトミックサービスでなければならない。かつ環境中に同名のサービス名によるサービス呼出し関数がなければならない。上記サービス定義マクロにおけるサービス選択手続きは下位サービスがない場合には、グラウンディング情報によって正しく引数の写像を行い、実際の Web サービスを呼び出すものとしている。

4. 今後の課題

本報告では Web サービスの分解・実行のために、CLOS の `funcallable-standard-class` のインスタンスとしての Web サービスとその解釈実行の機構を実装する提案をした。

ここではセマンティック Web 用プロセッサ SWCLOS の利用を前提としている。SWCLOS では CLOS オブジェクトと OWL が融合されているが、今後は関数呼出し可能な Web サービス処理と階層型計画手法を開発し、実用的 Web サービスの分解・実行システムを開発する。Web サービスの合成にはこれとは別に、部分順序計画機能を有する Web サービス合成システムを開発する予定である。

参考文献

- [Koide] Koide, S., Aasman, J., and Haflich, S.: OWL vs. Object Oriented Programming, Workshop on Semantic Web Enabled Software Engineering (SWESE) at the 4th International Semantic Web Conference (ISWC2005) <http://www.mel.nist.gov/msid/conferences/SWESE/repository/8owl-vs-OOP.pdf> (2005).
- [Nau] Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D. and Yaman, F.: SHOP2: An HTN Planning System, Journal of Artificial Intelligence Research, Vol. 20, pp.279-404 (2003).
- [小出 2005b] 小出: セマンティック Web サービス用エージェント - ロケット打上運用支援システムへの応用 -, 人工知能学会誌, Vol. 20, No. 6, pp.658-665, (2005).