

プロジェクトマネジメントへの Semantic Web サービスの適用の試み

中挾知延子

東洋大学国際地域学部 〒374-0193 群馬県邑楽郡板倉町泉野 1-1-1

E-mail: chiekon@toyonet.toyo.ac.jp

あらまし 本稿ではプロジェクトマネジメントにおいて種々のプロセスが並行して行われるときに、各プロセスを担う組織間での情報交換の手段として OWL を用いた Semantic Web Services を提案する。OWL-S と SWRL および WSDL を連携させた Web サービスにより、今までプロジェクトマネジメントにおいて失敗した場合の一因として考えられる、組織間での情報交換不足の解決に向けての 1 つの試みとしたい。

キーワード OWL, OWL-S, Semantic Web サービス, SWRL, プロジェクトマネジメント

An Attempt for Applying Semantic Web Services to Project Management

Chieko NAKABASAMI

Faculty of Regional Development Studies, Toyo University

1-1-1 Izumino, Itakura-machi, Oura, Gunma 374-0193 Japan

E-mail: chiekon@toyonet.toyo.ac.jp

Abstract In this paper, we report an attempt to use ontology-aware semantic web services as a means for information exchange among multiple organizations, each of which is responsible for a process which is performed in parallel with other processes as part of a project. By incorporating OWL-S, SWRL, and WSDL, we expect to solve the 'lack of information problem' which is thought to cause project management failure.

Keyword OWL, OWL-S, Semantic Web Services, SWRL, Project Management

1. はじめに

プロジェクトマネジメントとは、チームに与えられた目標を達成するために、人材・資金・設備・物資・スケジュールなどをバランスよく調整し、全体の進捗状況を管理する手法のことである。本稿ではプロジェクトマネジメントにおいて種々のプロセスが並行して行われるときに、各プロセスを担う組織間での情報交換の手段として Semantic Web サービス[1]を提案する。具体的な実現方法として OWL-S[2]と SWRL[3]および WSDL[4]を連携させることを試みている。本手法により、今までプロジェクトマネジメントにおいて失敗した場合の一因として考えられる、組織間での情報交換不足の解決に向けての 1 つの試みとしたい。情報の管理に主眼をおいたプロジェクトマネジメントは、プロジェクトを構成するメンバーあるいは組織の持つ知識を表出化する[5]という意味では、組織知のナレッジマネジメントの 1 つである。

本稿では大規模なプロジェクトマネジメントにおけるナレッジマネジメントに焦点をあてている。大規

模なプロジェクトになると、異なった組織が多数関与し、それらの間での必要かつ有用な情報交換がプロジェクト成功への一因¹となる。各組織におけるプロジェクト遂行のための技術力は十分あるにもかかわらず、担当者同士の情報交換の怠慢が失敗を導いてしまう。とはいえ、適切なタイミングで必要な情報を受け取れるように日常配慮しておくことは仕事の最中には難しいことである。

そこで、Web サービスを用いてインターネットにアクセスするようにしておきさえすれば、各組織間で必要な情報をやりとりできる環境が実現できると考えた。プロジェクトにおける 1 つ 1 つの情報交換をプロセスととらえ、OWL-S でのプロセスオントロジによって複雑にからみあったプロセスを管理する。さらにプロセス同士の入出力における各情報が正しい両者で交換さ

¹ 専門家によると、近年起こった種々の事件の原因はプロジェクトの内部で、適切なときに必要な情報を各組織間で交換していなかったためだという。

れるように RuleML[6]と OWL[7]を結びつけた SWRL によって変数が扱えるルールでプロセスのより信頼性の高い管理ができると考えた。なお、適用については現在実験段階である。

以降の章構成を述べる。2章ではプロジェクトマネジメントの知識体系である PMBOK とその中の Project Communications Management について概説する。3章では PMBOK における Project Communications Management を基にプロセスに重点をおいたオントロジ記述を説明して、それらのオントロジに変数付きのルールを加えた記述を示す。4章で関連する技術ならびに今後の予定を述べて結びとする。

2. PMBOK によるプロジェクトマネジメントの知識体系

2.1. PMBOK の概要

プロジェクトマネジメントにおける情報の管理を考えるにあたって本研究では Project Management Body of Knowledge (PMBOK、ピンボックと呼ばれる) [8] をベースにしている。PMBOK は米国プロジェクトマネジメント協会[9]が提唱するプロジェクトマネジメントのための標準的な知識体系で、プロジェクトを統合的に管理するために以下の9つの知識領域をあげている。

- (1) Project Integration Management
- (2) Project Scope Management
- (3) Project Time Management
- (4) Project Cost Management
- (5) Project Quality Management
- (6) Project Human Resource Management
- (7) Project Communications Management
- (8) Project Risk Management
- (9) Project Procurement Management

PMBOK では (2)から(9)における8つの領域を統合するのが(1)の Project Integration Management であり、マネジメントしながら計画立案・実施していくことで、8つの領域が相互作用しながらバランスを取り、QCD (Quality・Cost・Delivery) を保証するものとされている。とくに(7)の Project Communications Management(以降 PCM と呼ぶ) では、適切なタイミングにおけるプロジェクト情報の生成・収集・公開・蓄積・配置について詳細なフレームワークが記されている。9つの領域はそれぞれいくつかの必要なプロセスで構成されており、さらにそれら各プロセスには *Inputs* と *Outputs* としてドキュメントまたはドキュメント可能な実体と、*Input* に適用され *Outputs* を作成する上で適用される *Tools and Techniques* が記されている。各プロセスは *Inputs* を受け取って機能を果たし、*Outputs* を結果とし

て別のプロセスあるいは領域に渡すことになる。

さらに PMBOK では、プロジェクトでの成果物の観点からプロセスグループという概念を設けている。プロセスグループは5つに分けられ、プロジェクトのライフサイクルを示しているといえる。先に述べた9つの領域における各プロセスはこれら5つのプロセスグループのいずれかにマッピングされている。ここで述べた PMBOK の概要について[8]からの抜粋を以下に図1、図2、表1で示す。9つの領域の例として PCM をあげ、そこにおける各プロセスがどのように5つのプロセスグループにマッピングされているかを示す。

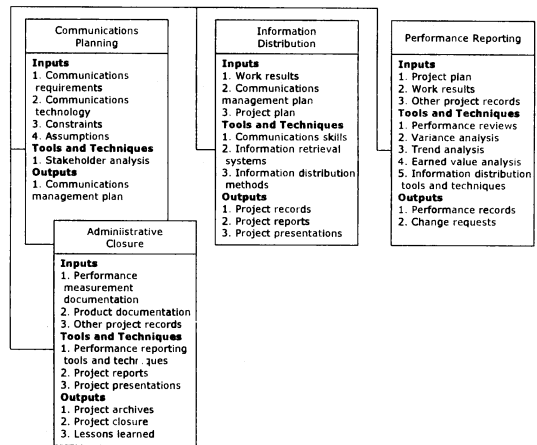


図1 PCMにおけるプロセス群

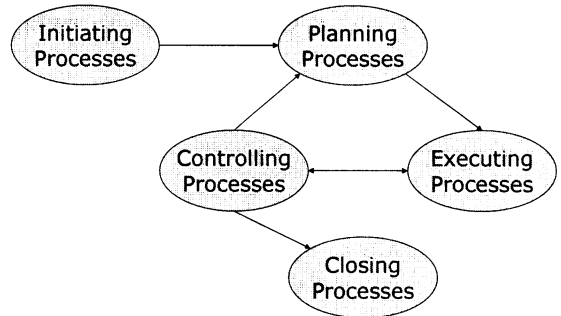


図2 5つのプロセスグループ

表1 プロセスグループとPCMのプロセスとのマッピング

Initiating	Planning	Executing	Controlling	Closing
	Communications Planning	Information Distribution	Performance Reporting	Administrative Closure

2.2. PCM の役割

PCM はプロジェクトを通じての組織知を管理する役割を果たす。2.1 節であげた PCM 以外の(2)から(9)の領域における各プロセスで発生する Outputs の情報を適切な別のプロセスへの Inputs として配布することである。本研究では PCM とそれ以外の領域を分けて考え、それ以外の領域における組織知を PCM によって管理することを考えている。例としてプロセスグループの 1 つである Executing Processes における各プロセスの関係を考える。文献[8]では図 3 に示すような関係が示されている。

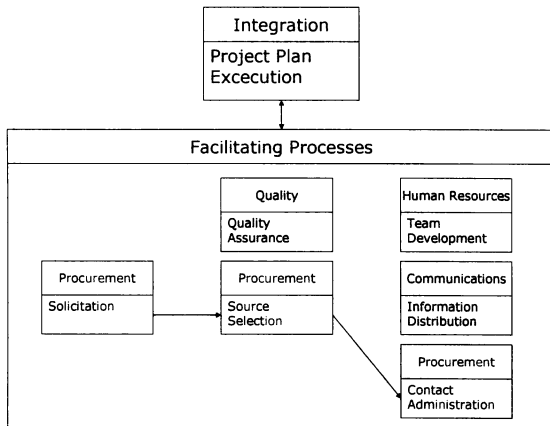


図 3 Executing Processes 中のプロセス

図 3 の Executing Processes における Facilitating Processes の内部を考えた場合、Communications 以外の Quality, Procurement, Human Resources の各プロセスにおける Inputs, Outputs の発生をそれらの内容とともにプロジェクト情報としてそれらに関わっている領域にアナウンスすることが必要であると考えられる。PCM では Web 上でのナレッジポータルサイトなどを通じて、Information Distribution プロセスにおける Inputs と Outputs をコンテンツとして関連組織へ伝えることになる。このとき Inputs および Outputs は図 3 における Communication 以外のプロセスより発生する情報である。

伝える手段として、サイトにアクセスしたとき必要な情報を受け取れるような一例として SAAJ[10]を使った Web サービスを考えている。そして、このような Web サービスの場合、受け取ったり送ったりする情報が意味することについての一貫性の保持と、1 つのプロセスにおいて必要な要素の関係ならびにプロセス間

での要素の受け渡しが行われるためにオントロジを取り入れることが有効であると考えられる。

3. OWL-S によるプロセス記述

OWL-S で提案されているプロセスオントロジでは、プロセスは Atomic, Simple, Composite に分類され、Atomic プロセスは Simple プロセスとして抽象的にグルーピングされ、実装にしたがって Composite プロセスとして組み立てられる[11]。

そこで、PMBOK をベースにしたプロセス記述として 9 領域を Simple、5 フェーズを Composite、子プロセスを Atomic とし、OWL-S の Process クラスのプロパティ hasInput, hasParticipant, hasOutput にそれぞれ Atomic プロセスで示されている Input, Tools and Techniques, Output を対応させたオントロジを作った。以下に作成したオントロジの一部を示す。

3.1. SWRL によるプロセス間ルール記述

リスト 1、2 で示したプロセス記述に、SWRL によるルール記述を加えて WSDL にリンクさせることにより Web サービスの内容を記述する。SWRL におけるルール記述のコア言語である RuleML は 0 個以上の前提と帰結を持つルールを XML ないし RDF で表現したものであり、一階述語論理のサブセットである DataLog と対応している。

OWL-S にもプロセス間の順序やループ、制約を表す要素が用意されているが、プロセス間で授受されるデータについて変数が使えない。そのため、複数のプロセスが複雑に絡み合う Web サービスなどでは、あるプロセスの入力がどのプロセスの出力を参照しているか、2 つのプロセスの持つプロパティ値が同一のリソースを参照しているかなどが表現できない。プロジェクトマネジメントにおける複数のプロセスで行き交う情報を正確に扱うには SWRL もとりこむ必要があると考えた。文献[12]を参考にした記述の一部をリスト 3 に示す。オントロジ記述である OWL ファイルと WSDL ファイルの関係を図 4 に示す。プロジェクトマネジメントのオントロジにおけるプロセスの定義が、WSDL におけるメソッド記述ともいえる operation エレメントの 1 つの属性として記述される。

リスト1 プロジェクトマネジメントにおけるプロセスオントロジの一部

```

<process:AtomicProcess rdf:ID="ScheduleDevelopment">
<process:hasOutput rdf:resource="#ProjectSchedule_Out" /></process:AtomicProcess>
<process:UnConditionalOutput rdf:ID="ProjectSchedule_Out">
<process:parameterType rdf:resource=http://chieko.com/pm/Concepts.owl#Schedule/>
</process:UnConditionalOutput>
<process:AtomicProcess rdf:ID="ScheduleControl">
<process:hasInput rdf:resource="#ProjectSchedule_In" /></process:AtomicProcess>
<process:Input rdf:ID="ProjectSchedule_In">
<process:parameterType rdf:resource=http://chieko.com/pm/Concepts.owl#Schedule/>
</process:Input>
<process:SimpleProcess rdf:ID="ProjectTimeManagement">
<process:expandTo><process:composedOf><process:Iterate>
<process:components rdf:parseType="Collection">
<process:AtomicProcess rdf:about="#ActivityDefinition" />
<process:AtomicProcess rdf:about="#ActivitySequencing" />
<process:AtomicProcess rdf:about="#ActivityDurationEstimating" />
<process:AtomicProcess rdf:about="#ScheduleDevelopment" />
<process:AtomicProcess rdf:about="#ScheduleControl" />
</process:components></process:Iterate></process:composedOf></process:expandTo>
</process:SimpleProcess>
<process:CompositeProcess rdf:ID="Controlling"><process:composedOf><process:Unordered>
<process:components rdf:parseType="Collection">
<process:AtomicProcess rdf:about="#IntegratedChangeControl" />
<process:AtomicProcess rdf:about="#ScopeVerification" />
<process:AtomicProcess rdf:about="#ScopeChangeControl" />
<process:AtomicProcess rdf:about="#ScheduleControl" />
<process:AtomicProcess rdf:about="#CostControl" />
<process:AtomicProcess rdf:about="#QualityControl" />
<process:AtomicProcess rdf:about="#PerformanceReporting" />
<process:AtomicProcess rdf:about="#RiskMonitoringAndControl" />
</process:components></process:Unordered></process:composedOf></process:CompositeProcess>

```

リスト2 Concept.owlの一部

```

<!-- Inputs and Output -->
<owl:Class rdf:ID="HistoricalInformation"/><owl:Class rdf:ID="OrganizationalPolicies"/>
<owl:Class rdf:ID="Constraints"/><owl:Class rdf:ID="Assumptions"/>
<owl:Class rdf:ID="ProjectPlan"/><owl:Class rdf:ID="SupportingDetail"/>
<owl:Class rdf:ID="CorrectiveAction"/><owl:Class rdf:ID="Results"/>
<owl:Class rdf:ID="Requests"/><owl:Class rdf:ID="Reports"/>
<owl:Class rdf:ID="Lessons"/><owl:Class rdf:ID="Description"/>
<owl:Class rdf:ID="Criteria"/><owl:Class rdf:ID="Charter"/>
<owl:Class rdf:ID="Statement"/><owl:Class rdf:ID="Structure"/>
<owl:Class rdf:ID="Documentation"/><owl:Class rdf:ID="Acceptance"/>
<owl:Class rdf:ID="Change"/><owl:Class rdf:ID="ActivityList"/>
<owl:Class rdf:ID="Dependency"/><owl:Class rdf:ID="Diagram"/>
<owl:Class rdf:ID="Requirement"/><owl:Class rdf:ID="Capability"/>
<owl:Class rdf:ID="Estimate"/><owl:Class rdf:ID="Calendar"/>
<owl:Class rdf:ID="Schedule"/>
<!-- Tools and Techniques -->
<owl:Class rdf:ID="PMethodology"/><owl:Class rdf:ID="PMSkillsAndKnowledge"/>
<owl:Class rdf:ID="PMSystem"/><owl:Class rdf:ID="PMMeetings"/>
<owl:Class rdf:ID="PMProcedures"/><owl:Class rdf:ID="PMConfiguration"/>
<owl:Class rdf:ID="PMMeasurement"/><owl:Class rdf:ID="PMPlanning"/>
<owl:Class rdf:ID="PMJudgement"/><owl:Class rdf:ID="PMAnalysis"/>
<owl:Class rdf:ID="PMIdentification"/><owl:Class rdf:ID="PMStructureTemplates"/>
<owl:Class rdf:ID="PMDecomposition"/><owl:Class rdf:ID="PMInspection"/>
<owl:Class rdf:ID="PMControl"/><owl:Class rdf:ID="PMTemplates"/>
<owl:Class rdf:ID="PMEstimating"/><owl:Class rdf:ID="PMSimulation"/>
<owl:Class rdf:ID="PMScheduling"/><owl:Class rdf:ID="PMHeuristics">
<owl:Class rdf:ID="PMSoftware"/>

```

リスト3 SWRLによるルール記述の一部

```

<!-- SWRL notation -->
<!-- hasInput(P2, Info) :- ScheduleDevelopment(P1),
ProjectSchedule(Info),hasOutput(P1, Info),ScheduleControl(P2),ProjectSchedule(Info). -->
<ruleml:imp>
<ruleml:_body>
<swrl:classAtom>
<owl:Class owl:name="&process;ScheduleDevelopment"/>
<ruleml:var>scheduleDev</ruleml:var></swrl:classAtom>
<swrl:classAtom>
<owl:Class owl:name="&process;ProjectSchedule"/>
<ruleml:var>pschedule</ruleml:var></swrl:classAtom>
<swrl:classAtom>
<owl:Class owl:name="&process;ScheduleControl"/>
<ruleml:var>scheduleCont</ruleml:var></swrl:classAtom>
<swrl:individualPropertyAtom swrl:property="&process;hasOutput">
<ruleml:var>scheduleDev</ruleml:var>
<ruleml:var>pschedule</ruleml:var>
</swrl:individualPropertyAtom></ruleml:_body>
<ruleml:_head>
<swrl:individualPropertyAtom swrl:property="&process;hasInput">
<ruleml:var>scheduleCont</ruleml:var>
<ruleml:var>pschedule</ruleml:var>
</swrl:individualPropertyAtom></ruleml:_head>
</ruleml:imp>
<!-- Announce(PG1,PG2,Info) :-
ProjectIntegrationManagement(PG1),ProjectTimeManagement(PG2),ScheduleUpdates(Info),hasOutput(PG
2,Info). -->
<ruleml:imp>
<ruleml:_body>
<swrl:classAtom>
<owl:Class owl:name="&process;ProjectIntegrationManagement"/>
<ruleml:var>pintegration</ruleml:var></swrl:classAtom>
<swrl:classAtom>
<owl:Class owl:name="&process;ProjectTimeManagement"/>
<ruleml:var>ptime</ruleml:var></swrl:classAtom>
<swrl:classAtom>
<owl:Class owl:name="&process;ScheduleUpdates"/>
<ruleml:var>scheduleUp</ruleml:var></swrl:classAtom>
<swrl:individualPropertyAtom swrl:property="&process;hasOutput">
<ruleml:var>ptime</ruleml:var>
<ruleml:var>scheduleUp</ruleml:var></swrl:individualPropertyAtom></ruleml:_body>
<ruleml:_head>
<swrl:individualPropertyAtom swrl:property="Announce">
<ruleml:var>ptime</ruleml:var>
<ruleml:var>pintegration</ruleml:var>
<ruleml:var>scheduleUp</ruleml:var></swrl:individualPropertyAtom></ruleml:_head>
</ruleml:imp>

```

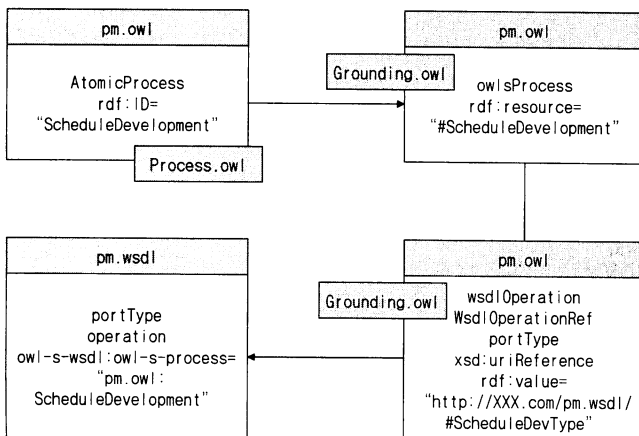


図4 OWLとWSDLの関係

4. 考察と今後の予定

本研究報告では、プロジェクトマネジメントにおける情報交換の重要性に着目し、実現手法の1つとしてオントロジを適用した Semantic Web Services からのアプローチを示した。Semantic Web Services のプロジェクトマネジメントへの適用については現在実験中であり十分な考察が得られるまでには至っていない。実装方法として OWL-S、SWRL などを試みており、これらが WSDL にリンクされることにより処理は可能である。RuleML をコアとする SWRL についての解析エンジンの充実が必要である。

概して Web サービスが現実に広く使われるためには、異質な複数の Web サービスが矛盾なく統合されなければならない。そのためには Web サービスにおけるインタフェースにオントロジでの意味概念がリンクされた形で提供されることが望ましい[13]。また Web サービスが合成されたものは Web プロセスの一部として扱われるべきであり、それら Web プロセス間の複雑な制御がルール記述と推論によって解決できると考えられる。最近、Web サービスの連携におけるプロセスフローやメッセージ交換の詳細をとりあげたコレオグラフィ記述言語が登場している[14]。OWL-S では変数が扱えないのに対して、SWRL ならびにこれらのコレオグラフィ記述言語では扱えることも、Web プロセスをより詳細に記述できる利点である。今後も Web サービスと Web オントロジに着目して、プロジェクトマネジメントでの適用と情報交換のシステム実現を進めていく予定である。

文献及び参照 URL

- [1] Semantic Web Services Interest Group
<http://www.w3.org/2002/ws/swsig/>
- [2] DAML Services.<http://www.daml.org/services/owl-s/>
- [3] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2003
<http://www.daml.org/2003/11/swrl/rules-all.html>
- [4] Web Services Description Language (WSDL) 1.1.
<http://www.w3.org/TR/wsd1>
- [5] 野中郁次郎、竹内弘高、梅本勝弘(訳), "知識創造企業", 東洋経済新報社, 1996
- [6] RuleML Homepage, <http://www.ruleml.org/>
- [7] Web-Ontology (WebOnt) Working Group
<http://www.w3.org/2001/sw/WebOnt/>
- [8] Project Management Institute, "A Guide to the Project Management Body of Knowledge (PMBOK Guide) 2000 Edition", Project Management Institute, 2000
- [9] Project Management Institute (PMI) Home Page.
<http://www.pmi.org/info/>
- [10] SOAP with Attachments API for Java (SAAJ),

<http://java.sun.com/xml/saaj/>

- [11] The OWL Services Coalition, "OWL-S: Semantic Markup for Web Services", 2003
<http://www.daml.org/services/owl-s/1.0/owl-s.html>
- [12] David Martin, Mark Burstein, Ora Lassila, Massimo Paolucci, Terry Payne, Sheila McIlraith, "Describing Web Services using OWL-S and WSDL", DAML-S Coalition working document, 2003
<http://www.daml.org/services/owl-s/1.0/owl-s-wsdl.html>
- [13] Jorge Cardoso, Christoph Bussler, Amit Sheth and Dieter Fensel, "Semantic Web Services and Processes: Semantic Composition and Quality of Service", PowerPoint for Tutorial at Federated Conferences On the Move to Meaningful Internet Computing and Ubiquitous Computer 2002, Irvine CA, October 2002.
- [14] 服部和彦, "Web サービスを連携させるコレオグラフィ", @IT:連載 ビジネス Web サービス最新事情 (5), 2003.
<http://www.atmarkit.co.jp/fxml/tanpatsu/25websvc/05.html>