

# Swashup

*The RoR approach to the SWS Challenge*

**E. Michael Maximilien, Ph.D.**

<http://maximilien.org>

IBM Almaden Research Center

San Jose, CA

v0.6.0

# About me

- Research Staff Member
- Almaden Services Research group
- Silicon valley, CA
- **NOTE** information and positions in this presentation represent my ideas and research architecture, design, and implementation, and may not be official positions of my Research employees, in the IBM, Corp. or its services, mashups, OOD/P, SE, and Agile methods
- Ph.D. in computer science from NC State University, Raleigh





First they ignore you...



next they laugh at you...

then you persevere, you win, and you fly...

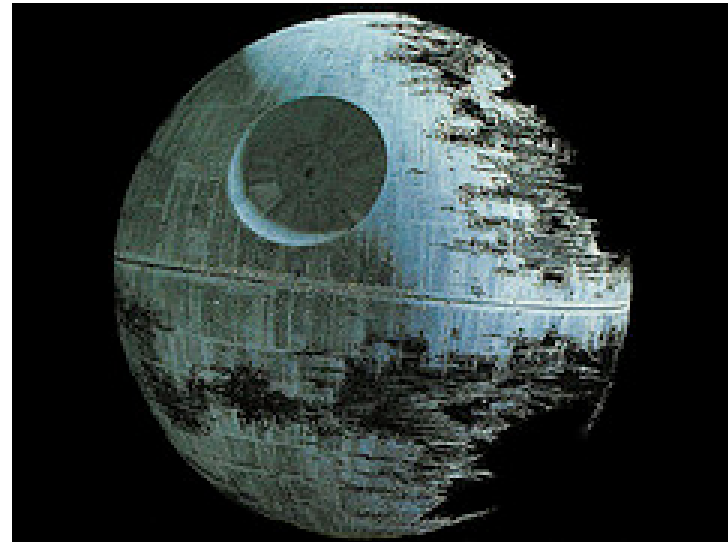
# Agenda

- Opinion
- Trends
- Questions and definitions
- Problems and solution
- Challenge
- Status
- Comparison
- Directions

# Opinion

Millennium push for “enterprisy” Web software led to...

**WS -**



credit [“David H. Hanson”, “LucasFilm Ltd.”, “<http://flickr.com/photos/rag/274029595/>”]

# Trends

- **Social , participation, and programmable Web**
- **Services era** (post-Web)
  - ▶ Programmable informational services
- Era of loose, UI-based, heterogeneous, and situational
  - ▶ Remixing of applications and data service composition or **mashups**
  - ▶ Informal and *ad hoc* or *situational*
- **Web time** development require
  - ▶ Agile programming
  - ▶ Scripting-based languages and platforms

# Questions?

- **Mashups**
  - ▶ data integration?
  - ▶ user-interface integration?
  - ▶ process integration?
- What about semantics?
- Ad hoc nature of mashups?
- What are mashups?

# Definitions - Mashups

*“Web application composed of multiple services with user-facing components”*

- Typically can be divided into
  - ▶ **data** mediation
  - ▶ **process** or protocol resolution
  - ▶ **user-interface** customization
- Various levels of mashups

# Definitions

- What are **Web services**?
  - ▶ Web-based API with defined XML input and output
  - ▶ SOAP and WSDL based Web APIs
- What are **Web APIs**?
  - ▶ XML data streams, e.g., Atom and RSS
  - ▶ XML Web functions using REST architectural style

# Problems

- **Web services**
  - ▶ too complex
  - ▶ not agile
  - ▶ missed Web 2.0 push
  - ▶ lack human interface
- **Semantic Web services** are *in the air*
- **Mashups** are ad hoc

# Solution

- Uniform agile platform for developing and managing Web 2.0 applications and mashups
- Common service models
- DSLs à go-go
  - ▶ tagging, commenting, digging, searching
  - ▶ service and mashup
- Tooling
- Secret sauce



**Ruby**  
*A Programmer's Best Friend*



# Why Ruby?

- Modern and dynamic OO programming language

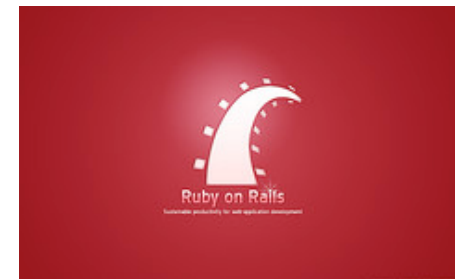
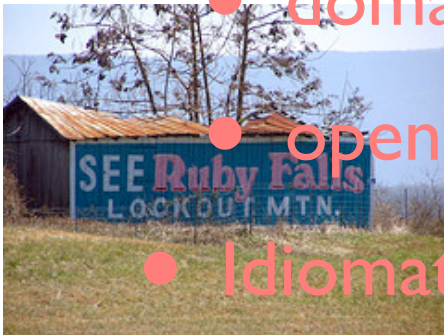
- Key features

- mixins, duck-typing

- domain-specific languages == DSLs

- open-source

- Idiomatic and *opinionated*



# Why Rails?

- Agile Web-based framework
- Philosophy
  - Convention over configurations
  - Don't Repeat Yourself (DRY)
  - Do it in Ruby!
  - Small tools
- Abstracted from real-world experiences



# RoR vs. Java

credit [["http://www.youtube.com/watch?v=PQbuyKUaKFo"](http://www.youtube.com/watch?v=PQbuyKUaKFo)]

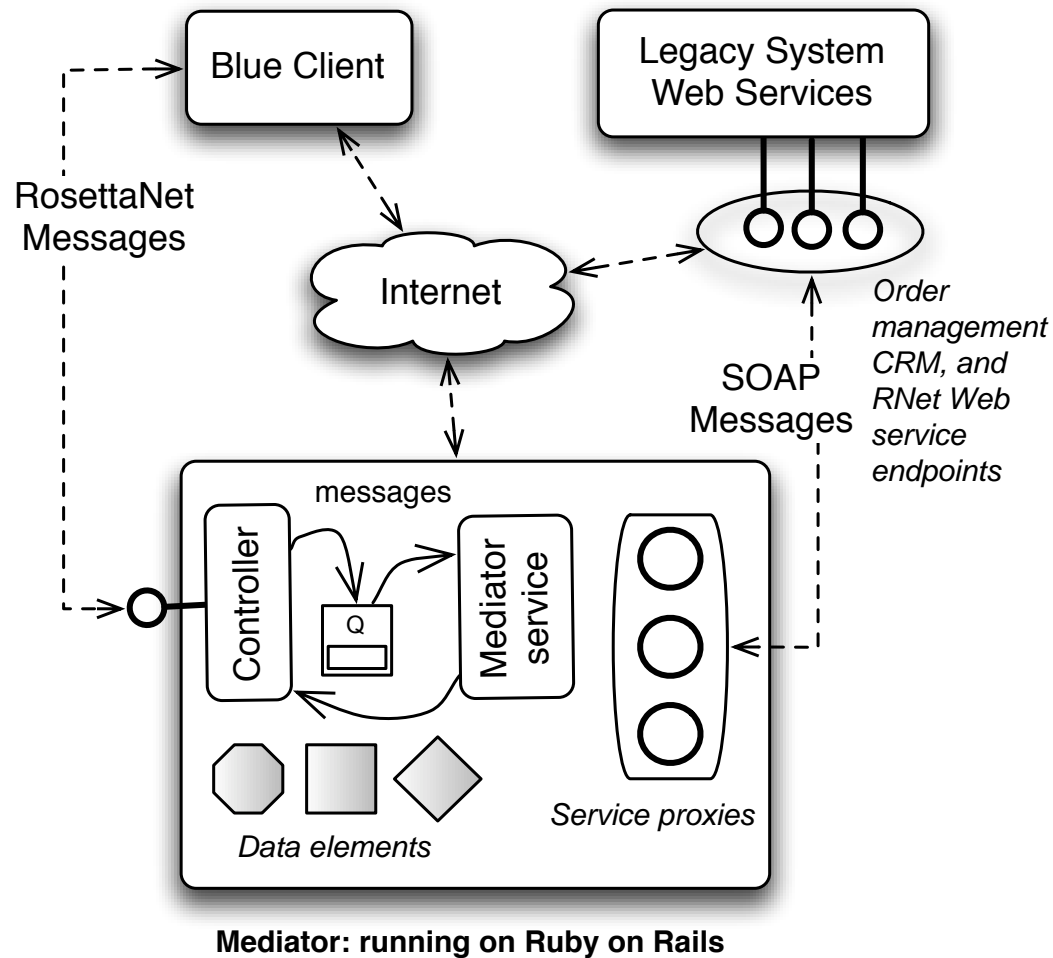
# Why DSLs?

- **Mini language** focused on some “domain”
  - ▶ built on top of another language
  - ▶ facilitate programming in the domain
- **Key values**
  - ▶ abstract
  - ▶ terse-code
  - ▶ code generation
  - ▶ natural syntax  $\Rightarrow$  ease of programming

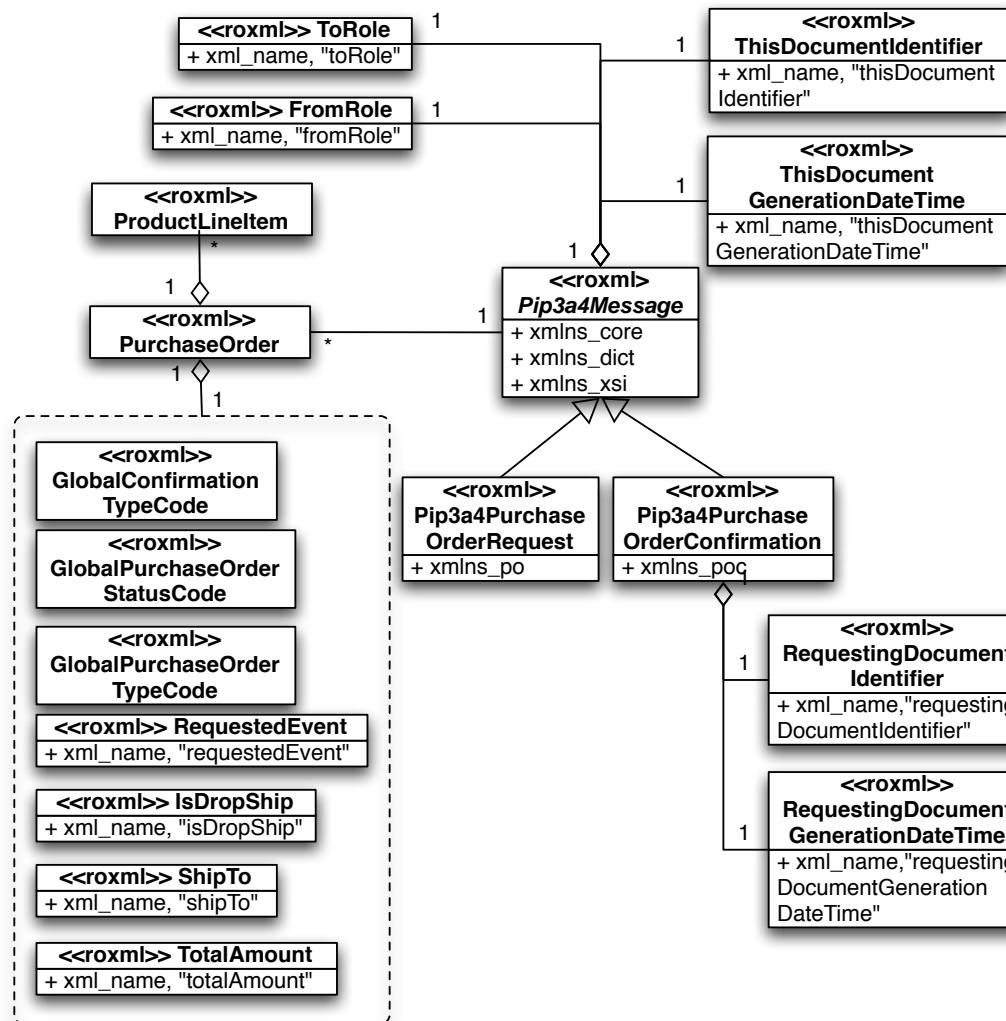
# Challenge

- Set of **non-trivial** problems
- Focuses on various aspects of integration
  - ▶ **data**
  - ▶ **process**
- Somewhat **realistic** and uses a real test-bed
- **Open source** and **collaborative**

# Solution - architecture sketch



# Solution - data models



# Solution - data model (cont.)

Listing 2: Pip3a4PurchaseOrderRequest model.

```

1 class Pip3a4PurchaseOrderRequest
2   include ROXML, Swashup::Creatable
3   xml_name 'po:Pip3a4PurchaseOrderRequest'

5   xml_attribute :xmlns_po, 'xmlns:po'
6   xml_attribute :xmlns_core, 'xmlns:core'
7   xml_attribute :xmlns_dict, 'xmlns:dict'

9   xml_object :fromRole, FromRole
10  xml_object :PurchaseOrder, PurchaseOrder
11  xml_object :thisDocumentGenerationDateTime,
12            ThisDocumentGenerationDateTime
13  xml_object :thisDocumentIdentifier,
14            ThisDocumentIdentifier
15  xml_object :toRole, ToRole
end

```

Listing 4: PurchaseOrder ROXML class.

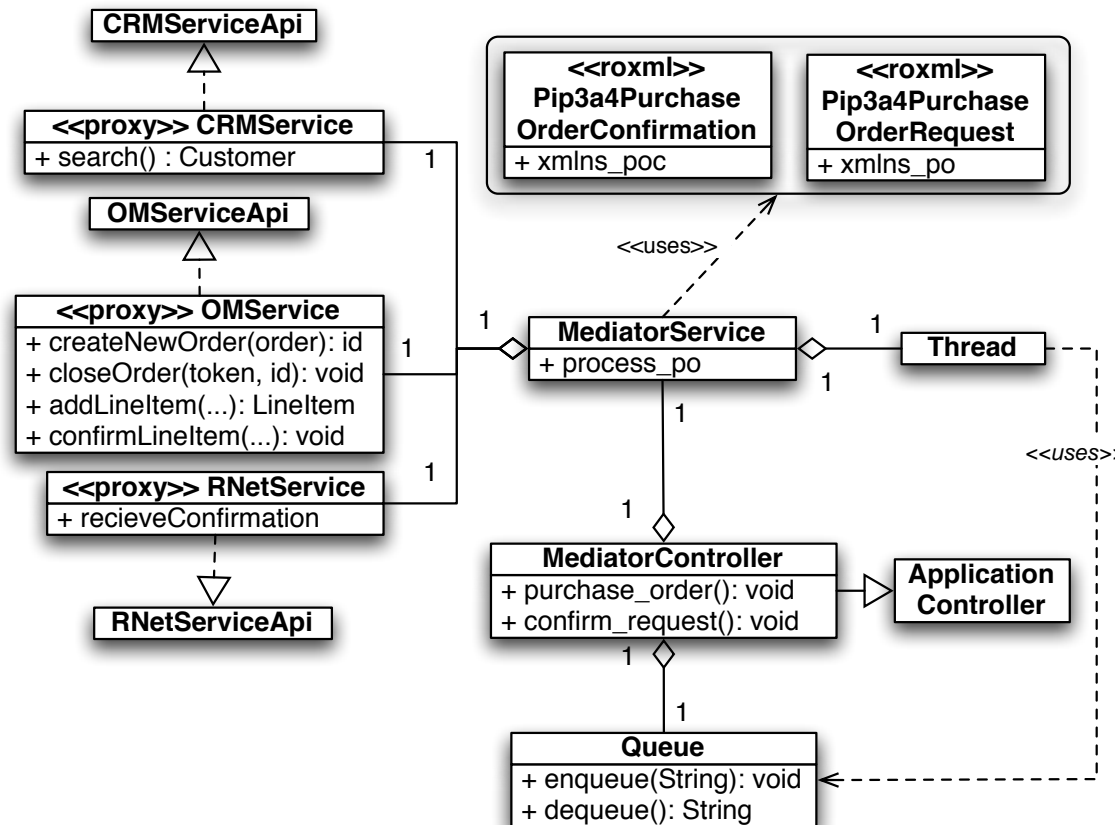
```

1 class PurchaseOrder
2   include ROXML, Swashup::Creatable
3   xml_name 'PurchaseOrder'
4   xml_text :GlobalConfirmationTypeCode,
5           'dict:GlobalConfirmationTypeCode'
6   xml_text :GlobalPurchaseOrderStatusCode,
7           'dict:GlobalPurchaseOrderStatusCode'
8   xml_text :GlobalPurchaseOrderTypeCode,
9           'dict:GlobalPurchaseOrderTypeCode'

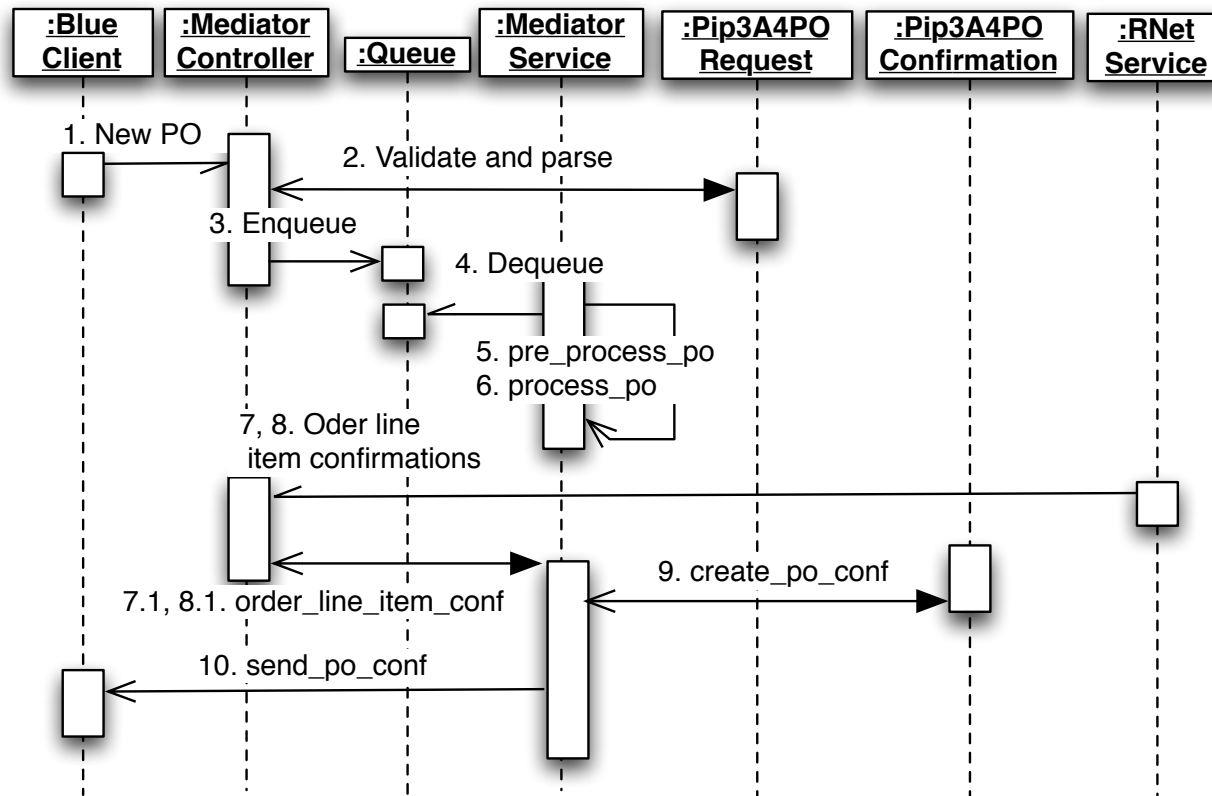
11  xml_object :isDropShip, IsDropShip
12  xml_object :ProductLineItem,
13            ProductLineItem
14  xml_object :requestedEvent,
15            RequestedEvent
16  xml_object :shipTo, ShipTo
17  xml_object :totalAmount, TotalAmount
end

```

# Solution - mediator architecture



# Solution - mediator process



# Solution - mediator (cont.)

```

class MediatorService
2  include Swashup, ServiceHelper
  # include many other modules
4
  def purchase_order po_request_message
6    request = Pip3a4PurchaseOrderRequest.
      parse(message)
8    pre-process request
      process request
10   end

12   def post_process request
      @order_info = nil
14     @conf = create_conf(request)
      submit_conf @conf
16     @ack_conf = create_ack_conf(@conf)
      [@conf, @ack_conf]
18   end

20   private

22   def pre_process request
      @order_info = OrderInfo.new
24     @line-item-infos = []
      validate_request request
26     @ack = create_ack_request(request)

```

```

28
  def process request
30   request.fromRole.
      PartnerRoleDescriptions.
32     each do |desc|
        pd = desc.PartnerDescription
34       customer = search_customer(pd)
        id = create_order(customer, desc)
36       request.PurchaseOrder.
          ProductLineItems.each do |pli|
38         add_line_item(id, pli)
          end
40       @om_service.
          close_order(@auth_token, id)
42     end
  end

44   def search_customer desc
46     text = desc.BusinessDescription.
          businessName.FreeFormText
48     @crm_service.search(text)
  end

50   def create_ack_conf conf
52     omd = OriginalMessageDigest.create
          :Pip3A4PurchaseOrderConfirmation
          => conf
54     nrm = NonRepudiationInformation.create
          :OriginalMessageDigest => omd
          ReceiptAcknowledgment.create
56     :NonRepudiationInformation
          => nrm
58   end
60 end

```

# Status

- Example applications catalog available
- W2 DSLs available as RoR plugins
- Mashup DSL soon available
- Tooling under development in summer
- Private beta mid-summer
- Public IBM-beta end of summer
- OSS version of some components (e.g., DSLs) end of summer

# Demo

- RoR Web application
- RosettaNet data using ROXML DSL
- Custom SOAP-client
- Mediator service
  - **In** endpoint (receive POs)
  - **Out** endpoint (submit confirmations)
- Solves mediation problems (only for now)

# Benefits

- Common architecture for all mashups
- Common unified service model
- Common primitives for Web 2.0 application construction
- Semantics are part of the data definition
- **Agility** and **simplicity**
- Toward a common platform for mashups and situational applications



# Comparison (partial)

- **Advantages**

- ▶ software engineering and programming
- ▶ agility
- ▶ small and terse code

- **Disadvantages**

- ▶ lack of maturity and tooling
- ▶ lacks means for representing business rules
- ▶ semantics are embedded (exchanged via DSLs)

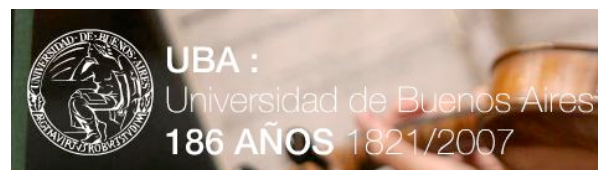
# Directions

- Other Web 2.0 applications
- **Complete** SWS Challenge problems
  - ▶ Mediation problems (80% done)
  - ▶ Discovery solution using Milano's disco engine as a service
- Move process model to Swashup service DSL
- End-user mashup **tooling**
  - ▶ Management of mashups and W2As

# Acks

## IBM

- Ajith Ranabahu
- Stefan Tai
- N.C. Narendra



**NC STATE UNIVERSITY**



# Acks multimedia

- ◆ <http://flickr.com/photos/encyte/56883622/>
- ◆ <http://flickr.com/photos/in-search-of-syd-isos/392809788/>
- ◆ <http://flickr.com/photos/gamany/287154266/>
- ◆ <http://flickr.com/photos/wapqueen/405898255/>
- ◆ <http://flickr.com/photos/ittybitty/508151822/>
- ◆ <http://flickr.com/photos/spastica/247377476/>
- ◆ <http://flickr.com/photos/netriver/405080963/>
- ◆ <http://flickr.com/photos/rooreynolds/349122523/>
- ◆ <http://flickr.com/photos/rag/274029595/>
- ◆ <http://flickr.com/photos/cknlomein/315262157/>
- ◆ <http://flickr.com/photos/tel/9168898/>
- ◆ <http://flickr.com/photos/automania/96820376/>
- ◆ <http://flickr.com/photos/areyarey/199837825/>
- ◆ <http://flickr.com/photos/txiribiton/237127315/>
- ◆ <http://flickr.com/photos/pingnews/290236914/>
- ◆ <http://flickr.com/photos/manahanwill/165766299/>
- ◆ <http://flickr.com/photos/adelig/448767224/>
- ◆ <http://flickr.com/photos/asmythie/180645843/>
- ◆ <http://www.youtube.com/watch?v=PQbuyKUaKFo>
- ◆ <http://www.youtube.com/watch?v=p5ElrSM8dCA>
- ◆ <http://www.youtube.com/watch?v=n1NVfDIU6yQ&NR=1>
- ◆ [Star Wars Death Star Picture from Lucas Film, Ltd.](#)
- ◆ [“The Empire March” from \*The Empire Strikes Back\* score by John Williams](#)
- ◆ [“Hotel California” by the \*Eagles\* and as interpreted by Toto Laraque](#)

# References

- E. M. Maximilien, H. Wilkinson, N. Desai, and S. Tai “A Domain-Specific Language for Web Services and APIs Mashups”, *to appear*
- E. M. Maximilien et al. “Swashup: Toward a Platform for Web 2.0 Applications and Mashups Development”, *to appear*
- E. M. Maximilien “A Partial Solution to the SWS Challenge: The Ruby on Rails Approach”, ICIES 2007 Workshop, *to appear*

धन्यवाद  
Hindi

多謝  
Traditional Chinese

ขอบพระคุณ  
Thai

Спасибо  
Russian

Gracias  
Spanish

شكراً  
Arabic

Thank You  
English

Obrigado  
Brazilian Portuguese

Grazie  
Italian

多谢  
Simplified Chinese

Danke  
German

Merci  
French

நன்றி  
Tamil

ありがとうございました  
Japanese

# Backup



# Solution (details) - mashup DSL grammar

```

EOL ::= '\n'
3 ALPHA ::= [a-zA-Z]
  ALPHA.NUMERIC ::= ALPHA + [0123456789]
5 VALID_SYMBOLS ::= ...

7 # Ruby's grammar (not complete)
  RUBY_TYPES ::= :string | :integer | :boolean
9 RUBY_STRING ::= ALPHA [ALPHA.NUMERIC|VALID_SYMBOLS]
  RUBY_SYMBOL ::= :RUBY_STRING
11 RUBY_STATEMENT ::= ...
  RUBY_HASH ::= {RUBY_SYMBOL|RUBY_STRING=>RUBY_STATEMENT}
13 VAR_NAMES ::= RUBY_STRING [, RUBY_STRING]+
  BLOCK_VARS ::= '[' VAR_NAMES '['

15
  BLOCK ::= do BLOCK_VARS
17         RUBY_STATEMENT+
         end

19
  # Swashup DSL grammar
21 TAG ::=
  tag RUBY_STRING
23 [, :synonyms => '[' [RUBY_STRING]+ '[' ]'?

25 TAGS ::= tags RUBY_STRING [, RUBY_STRING]+

27 VARS ::= {[RUBY_SYMBOL=>RUBY_TYPES|RUBY_SYMBOL]+}

```

```

29 MEMBER_STATEMENT ::=
  member RUBY_SYMBOL,
31 [RUBY_TYPES | RUBY_SYMBOL],
  [:xml_text | :xml_object]

33
  DATA ::= data RUBY_SYMBOL do
35         [TAG]?
         [TAGS]?
37         [xml_name RUBY_STRING]?
         [xml_attribute RUBY_SYMBOL]?
39         MEMBER_STATEMENT+
         end

41
  APLMETHOD ::= api_method RUBY_SYMBOL,
43         :expects => VARS,
         :returns => VARS

45
  API ::=
47 api RUBY_SYMBOL do
         [TAG]?
49         [TAGS]?
         [APLMETHOD EOL]+
51 end

53 SERVICE_OPTIONS ::=

```

# Solution - mashup DSL grammar (cont.)

```

[: alias => RUBY_SYMBOL,]?
55 [: wsdl => RUBY_STRING,]?
   :type => [: soap | :rest | :rss | :atom]+,
57 :api => RUBY_SYMBOL,
   :endpoint => RUBY_STRING
59
SERVICE ::= service RUBY_SYMBOL, SERVICE_OPTIONS
61
MEDIATION ::=
63 mediation RUBY_SYMBOL [ ,RUBY_SYMBOL]+ do BLOCK_VARS?
ENT) [TAG]?
65 [TAGS]?
   [RUBY_STATEMENT]+
67 end

69 MEDIATION_CALL ::=
   mediate :data, RUBY_SYMBOL, VAR_NAMES
71
OPERATION_WIRING ::=
73 wiring RUBY_SYMBOL, :operation [ ,:async]? do BLOCK_VARS?
   [[RUBY_NAME=?] MEDIATION_CALL]+
75 RUBY_STATEMENT+
   end
77
PROTOCOL_STEP ::=
79 step RUBY_SYMBOL do BLOCK_VARS?
   [TAG]?
81 [TAGS]?
   [[RUBY_NAME=?] [OPERATION_WIRING]]+
83 end

```

```

85 EXPOSE_SERVICE ::=
   expose_service SERVICE_TYPE,
87 :expects => VARS,
   :returns => VARS,
89 [: entry => RUBY_SYMBOL,]?
   [: metadata => RUBY_HASH,]?
91
PROTOCOL_WIRING ::=
93 wiring RUBY_SYMBOL, :protocol do BLOCK_VARS?
   [TAG]?
95 [TAGS]?
   [EXPOSE_SERVICE]?
97 [[RUBY_NAME=?] PROTOCOL_STEP]+
   end
99
MASHUP ::=
101 mashup RUBY_SYMBOL do BLOCK_VARS?
   [TAG]?
103 [TAGS]?
   [PROTOCOL_WIRING]+
105 end

```



# RoR vs. PHP

credit [<http://www.youtube.com/watch?v=p5ElrSM8dCA>]

# RoR vs. PHP (2)

credit [<http://www.youtube.com/watch?v=nINvfDIU6yQ>]