AN INTERNET OF RULES

SYSTEM OVERVIEW

Joseph Potvin
Executive Director, Xalgorithms Foundation
Some Questions on My Mind

General Framework:
• What conceptual map can illustrate RuleML vis-à-vis IoR functions?
• How might complementarity between IoR and RuleML be described?

Synergies:
• Can an IoR expedite the achievement of RuleML community goals?
• What challenges does RuleML face that an IoR may help to overcome?

Issues/Incongruities:
• Does any aspect of this overview seem incongruous with RuleML?
• Are there any apparent conceptual/technical/practical errors-omissions?

Experimentation:
• What exploratory projects might be useful over the next year?
• What challenges would the RuleML community put to IoR designers?
1. Principles
2. Problem Statement
3. XATP
4. Xalgo
5. Interlibr
6. Lichen
7. Sample Rules
8. Who Rules?
Internet Principles (IETF)

The Internet Principle: "Connectivity is its own reward"

smart edges, simple core

The Simplicity Principle: "Complexity is the primary mechanism which impedes efficient scaling"

seek the simplest possible solution

Occam's razor

"Plurality should not be posited without necessity."

The Least Power Principle: "The less powerful the language, the more you can do with the data stored in that language."

use concise declarative expressions so anyone can write programs for them

https://www.w3.org/2001/tag/doc/leastPower.html
https://www.w3.org/DesignIssues/Principles.html
Web Principles (W3C)

The Coupling Principle: "As things get larger, components exhibit increased interdependence."

loosely coupled systems

flexible time, sequencing, assumptions

https://www.w3.org/DesignIssues/Principles.html
Free/Libre Software Principles (FSF)

The Free Software Definition

Freedom 0:
Freedom to run the program for any purpose.

Freedom 1:
Freedom to study how the program works, and adapt it to one’s needs.

Freedom 2:
Freedom to copy and redistribute the program

Freedom 3:
Freedom to improve the program, and release any modified versions.
System Design Implications

A Rule Engine
Entire rulebase expressed and interpreted via one standard universal algorithm for systematic execution.

A Rule Fabric
Rules of a rulebase expressed in standard semantics via their own algorithms for customized execution.
‘Algorithms’ Implement ‘Rules’

**rule**  A guide to repeated behaviour by authority, agreement or preference.

**algorithm**  An operational method invoked by a specified data input condition to return a specified data output result, and then to terminate.
RULE: A normative precept by which repeated behaviour is guided through authority, agreement or preference.


ALGORITHM: A posited reusable operational method invoked by a specified data input condition to generate a specified data output result, and then to terminate.


ALGORITHM: A posited reusable operational method invoked by a specified data input condition to generate a specified data output result, and then to terminate.

**IF x, THEN y**

- **0th Conditional**
  - Present Uncertain Fact
  - Present Certain Fact

- **1st Conditional**
  - Present Uncertain Situation
  - Future Certain Action

- **2nd Conditional**
  - Present or Future Uncertain Situation
  - Present or Future Certain Action

**WHEN x | y**

- Certain Fact
- Certain Fact
A Problem Statement

How may anyone discover and obtain practical knowledge of all the institutional rules that are in effect & applicable to their particular circumstances at a given time?
An "Internet of Rules"

- **Xalgo-fact**
  - api data
  - find rules

- **Xalgo-rule**
  - report findings
  - deliver findings

- **Lichen**
- **Interlibr**
- **Xalgo**
  - publish rules

---

**XALGO-RITHMS Foundation**
Computing Interoperability Standards
Enabling Cross-Platform Networked Applications

- Computer Standards
  - “Operating System” Layers
  - Client System
    - Application
  - Operating System
  - Kernel
  - Hardware
  - “Domain Driven Design” Layers
  - User Interface
  - Application
  - Domain
  - Infrastructure

- Software Standards
  - “TCP/IP Model” Layers
  - Network System
    - Application
    - HTML, XML, JSON
    - HTTPS, SMTP, FTP
    - Transport Layer (TCP)
    - Internet Layer (IP)
    - Network Access Layer

- Network Standards
  - “Operating System” Layers
  - Server System
    - Application
    - Operating System
    - Kernel
    - Hardware

- “Computing Interoperability Standards”
  - Enabling Cross-Platform Networked Applications

XALGORITHMS Foundation

15 of 40
An Internet of Rules Via 3 Auxiliary Components
Creating a Seamless Request-Response Service for Finding Algorithms that Implement “In-Effect” & “Applicable” Rules
Optional: External Algorithms Transfer Protocol
Internet of Rules via Two Performance Modes (RE: Speed & Security) [JSON over HTTPS] or [CBOR over XATP]

Autonomous Distributed Connectors to HTTPS

Autonomous Distributed Connectors to XATP

XAlgorithms Foundation


Lichen (Xalgo Author)
Integrated Development Environment (IDE)

Software System
- User Interface
- Application
- Domain
- Infrastructure

Integrated Development Environment
- file.xalgo-rule
- file.xalgo-table

XML→JSON XML→CBOR Converter
- file.ruleml
- file.json or file.cbor

RESTful API Connector

Any Repository Service

New / Update
Versioned

EXTERNAL Versioning in a Git Service

“OUGHT” DOCUMENTS

Deontic Logic: e.g. LegalRuleML
- file.xalgo-rule
- file.xalgo-table

XALGORITHMS Foundation
Lichen (Xalgo Messenger)
Optimize Pre-Event Data for High-Performance Request-Response

Software System
- User Interface
- Application
- Domain
- Infrastructure

Local Data in-Memory

XML\(\rightarrow\)JSON XML\(\rightarrow\)CBOR Converter
- xalgo-is.xml to .json or .cbor
- .json or .cbor to xalgo-ought.xml

RESTful API Connector

REQUEST xalgo-is
RESPONSE xalgo-ought

input-file.xml
output-file.xml

Any Transaction Solution (e.g. commerce)

MESSAGES “IS” find “OUGHT” DOCUMENTS

- Commerce Documents: Universal Business Language (UBL)
- Payment Messages: UNIversal Financial Industry (UNIFI) Messaging Scheme (ISO 20022)
- Other standard schemas

XALGORITHMS Foundation
Interlibr
Request-Response Service is a sort of “Algorithms Search Engine”
Interlibr
Request-Response Service is a sort of “Algorithms Search Engine”

---

### DataStream

**Log incoming xalgo-is**

**Kafka**

**Log outgoing xalgo-ought**

---

### DataMart

- **Columns**
- **Queries**
- **Cassandra**
- **Documents**
- **History**
- **MongoDB**

---

### Interpreter (Scala Reference Implementation*)

A `xalgo-is.json` request message arrives as a key-value pair from Kafka.

1st CYCLE: Jurisdiction and date/time 'envelope' data are used to filter for all 'in effect' `file.xalgo-rule` packages.

2nd CYCLE: Results filtered for all 'applicable' `file.xalgo-rule` packages.

3rd CYCLE: Results are tested with all required `file.xalgo-table` lookups.

Descriptive `file.xalgo-rule` data and test results pass back to Kafka to log a `xalgo-ought.json` response message.

* For optimization by implementer into any production environment.

---

**EXTERNAL Versioning in a Git Service**

**ALGORITHMS**

---

**Server System**

**Application**

---

**Lichen**

---

### Diagram

- **xalgo-is** Request
- **xalgo-ought** Response
Lichen (Results Interface)

Design Problem: How to Present Many Equally-Significant Categories?

Lichen’s dynamic categories-oriented results interface compels simplicity through a 4±1 design constraint.

Maximums
- 4±1 new categories
- 4±1 clicks deep
- 4±1 hierarchical levels

Several Display Integration Options

- 2 × 4 × 2 × 5 = 80 categories
- 5 × 2 × 5 × 2 = 100 categories
- 3 × 4 × 3 × 4 = 144 categories
- 3 × 5 × 3 × 5 = 225 categories


A Sample Rule

Québec. Section 2R3(a) de la Règlement d'application de la Loi concernant la taxe sur les carburants (chapter T-1, r. 1). Ministère de justice, Québec (MJQ). Recueil des lois et des règlements du Québec (RLRQ). “LégisQuébec”.

http://legisquebec.gouv.qc.ca/fr/ShowDoc/cr/T-1,%20r.%201
**Appendix “A”**

**CS: Computer Systems Group annual rates of pay (in dollars)**

---

**Table Legend**

- $) Effective December 22, 2013
- A) Effective December 22, 2014
- B) Effective December 22, 2015
- X) Restructure effective April 1, 2016 (CS-01 to CS-04)
- C) Effective December 22, 2016
- D) Effective December 22, 2017

**CS-01: annual rates of pay (in dollars)**

<table>
<thead>
<tr>
<th>Effective Date</th>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
<th>Step 4</th>
<th>Step 5</th>
<th>Step 6</th>
<th>Step 7</th>
<th>Step 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$) December 22, 2013</td>
<td>53611</td>
<td>55933</td>
<td>57573</td>
<td>59541</td>
<td>61508</td>
<td>63474</td>
<td>65439</td>
<td>69068</td>
</tr>
<tr>
<td>A) December 22, 2014</td>
<td>54281</td>
<td>56288</td>
<td>58293</td>
<td>60285</td>
<td>62277</td>
<td>64267</td>
<td>66257</td>
<td>69952</td>
</tr>
<tr>
<td>B) December 22, 2015</td>
<td>54960</td>
<td>56992</td>
<td>59022</td>
<td>61039</td>
<td>63055</td>
<td>65070</td>
<td>67085</td>
<td>70626</td>
</tr>
<tr>
<td>X) Restructure effective April 1, 2016</td>
<td>55510</td>
<td>57582</td>
<td>59612</td>
<td>61649</td>
<td>63686</td>
<td>65721</td>
<td>67756</td>
<td>71534</td>
</tr>
<tr>
<td>C) December 22, 2016</td>
<td>56204</td>
<td>58282</td>
<td>60357</td>
<td>62420</td>
<td>64482</td>
<td>66543</td>
<td>68603</td>
<td>72428</td>
</tr>
<tr>
<td>D) December 22, 2017</td>
<td>56907</td>
<td>59011</td>
<td>61111</td>
<td>63200</td>
<td>65286</td>
<td>67375</td>
<td>69461</td>
<td>73333</td>
</tr>
</tbody>
</table>
A Stamp Duty on Real Property Purchases, based on Identity and Asset Ownership

Additional Buyer's Stamp Duty (ABSD) on Real Property Purchases, Inland Revenue Authority of Singapore. https://www.iras.gov.sg/irashome/Other-Taxes/Stamp-Duty-for-Property/Working-out-your-Stamp-Duty/Buying-or-Acquiring-Property/What-is-the-Duty-that-I-Need-to-Pay-as-a-Buyer-or-Transferee-of-Residential-Property/Additional-Buyer-s-Stamp-Duty--ABSD-1

A Sample Rule

Rates and Computation

<table>
<thead>
<tr>
<th>Profile of Buyer</th>
<th>ABSD Rates from 8 Dec 2011 to 11 Jan 2013</th>
<th>ABSD Rates from 12 Jan 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singapore Citizens (SC)(^1) buying first residential property</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>SC(^1) buying second residential property</td>
<td>Not applicable</td>
<td>7%</td>
</tr>
<tr>
<td>SC(^1) buying third and subsequent residential property</td>
<td>3%</td>
<td>10%</td>
</tr>
<tr>
<td>Singapore Permanent Residents (SPR)(^1) buying first residential property</td>
<td>Not applicable</td>
<td>5%</td>
</tr>
<tr>
<td>SPR(^1) buying second and subsequent residential property</td>
<td>3%</td>
<td>10%</td>
</tr>
<tr>
<td>Foreigners (FR) and entities(^2) buying any residential property</td>
<td>10%</td>
<td>15%</td>
</tr>
</tbody>
</table>
Xalgo Target Design Principles

Accessible Algorithms

1: Algorithms are declarative.
2: Algorithms embody patterns.
3: Algorithms are published.
4: Algorithms are simple.
Xalgo Functions

**WHEN:** Specify the facts that invoke the given rule’s algorithm with certainty.

**REQUIRE:** Preload a specific table before executing this rule.

**ASSEMBLE:** Construct a dynamic in-memory table from preloaded, computed or provided document data.

**REFINE:** Combined FILTER, MAP and REDUCE process as follows:

- **FILTER:** Remove rows from the computation before applying MAP assignments.
- **MAP:** Update or add column keys on a per-row basis.
- **TAKE:** Prune rows from the table after modifications have occurred.
- **KEEP:** Retain the virtual table until the next modifying statement.
- **REVISE:** Specify a particular permanent change to the indicated data.

**ARRANGE:** Offers the ability to algorithmically change the structure of the table (sorting, ordering of row)
Quebec: Border Retail Gas Tax Reduction

WHEN envelope:type == 'invoice';
WHEN envelope:parties.supplier.industry.list_id == 'ISIC';
WHEN envelope:parties.supplier.industry.value == 'G4711';
WHEN item:classification.list_name == 'UNSPSC';
WHEN item:classification.value == '506505';
WHEN item:quantity.value > 0;
REQUIRE ca.qc.tax:supplier_distances:0.1.0

REQUIRE ca.qc.tax:reductions_by_distance:0.1.0

ASSEMBLE sellers_reductions
  COLUMNS FROM table:reductions_by_distance

qc-gas-tax/reductions_by_distance.json
[
  { "distance" : 20, "reduction" : 0.00 },
  { "distance" : 15, "reduction" : 0.02 },
  { "distance" : 10, "reduction" : 0.04 },
  { "distance" : 5, "reduction" : 0.06 },
  { "distance" : 0, "reduction" : 0.08 }
]
Quebec: Border Retail Gas Tax Reduction

WHEN envelope:type == 'invoice';
WHEN envelope:parties.supplier.industry.list_id == 'ISIC';
WHEN envelope:parties.supplier.industry.value == 'G4711';
WHEN item:classification.list_name == 'UNSPSC';
WHEN item:classification.value == '506505';
WHEN item:quantity.value > 0;
REQUIRE ca.qc.tax:supplier_distances:0.1.0

REQUIRE ca.qc.tax:reductions_by_distance:0.1.0

ASSEMBLE sellers_reductions
  COLUMNS FROM table:reductions_by_distance
  ...

Canada: CS Group 303 Annual Rates of Pay

WHEN envelope:type == 'payment_authorization';
WHEN envelope:parties.supplier.industry.list_id == 'ISIC';
WHEN envelope:parties.supplier.industry.value == 'S9420';
WHEN item:classification.list_name == 'UNSPSC';
WHEN item:classification.value == '81111***';
WHEN item:quantity.value > 0;
REQUIRE ca..payroll:cs-group303_base-pay_by_contract:0.1.0
REQUIRE ca..payroll:cs-group303_base-pay_by_years-service:0.1.0

ASSEMBLE employees_base-pay
  COLUMNS FROM table:cs-group303_base-pay_by_contract
  ...
Quebec: Border Retail Gas Tax Reduction

WHEN envelope:type == 'invoice';
WHEN envelope:parties.supplier.industry.list_id == 'ISIC';
WHEN envelope:parties.supplier.industry.value == 'G4711';
WHEN item:classification.list_name == 'UNSPSC';
WHEN item:classification.value == '506505';
WHEN item:quantity.value > 0;
REQUIRE ca.qc.tax:supplier_distances:0.1.0

REQUIRE ca.qc.tax:reductions_by_distance:0.1.0

ASSEMBLE sellers_reductions
  COLUMNS FROM table:reductions_by_distance
  ...

Singapore: Additional Buyers’ Stamp Duty

WHEN envelope:type == 'option-to-purchase';
WHEN envelope:parties.buyer.industry.list_id == 'ISIC';
WHEN envelope:parties.buyer.industry.value == 'L6810';
WHEN item:classification.list_name == 'UNSPSC';
WHEN item:classification.value == '80131600';
WHEN item:quantity.value > 0;
REQUIRE sg..tax:buyerProfiles:0.1.1;

REQUIRE sg..tax:rates_by_profile:0.1.1;

ASSEMBLE buyers_rates
  COLUMNS FROM table:rates_by_profile
  ...

...
Event Data Source: UBL Invoice, Any Platform

REAL-TIME DATA FROM AN IN-PROGRESS PURCHASE

- **Industry code** for retail fuel vendors
- **Industry group name**: Retail Sale of Automotive Fuel
- **Vendor name**: l'Essence Chez Bob
Round-Trip: Completed UBL Invoice, Any Platform

<cbns:BaseUnitMeasure unitCode="LTR">1</cbns:BaseUnitMeasure>
<cbns:PerUnitAmount currencyID="CAD">0.04</cbns:PerUnitAmount>
-<cans:TaxCategory>
  -<cans:TaxScheme>
    <cbns:ID>QUEBEC BORDER GAS TAX REDUCTION</cbns:ID>
    <cbns:Name>Québec Border Gas Tax Reduction</cbns:Name>
  </cans:TaxScheme>
</cans:TaxCategory>
</cans:TaxTotal>
-<cans:Item>
  <cbns:Description>Regular Gas</cbns:Description>
  -<cans:CommodityClassification>
    <cbns:ItemClassificationCode listName="UNSPSC">506505</cbns:ItemClassificationCode>
  </cans:CommodityClassification>
  -<cans:AdditionalItemProperty>
    <cbns:ID>UNSPSC-NAME</cbns:ID>
    <cbns:Name languageID="EN">Gasoline and Petrol</cbns:Name>
  </cans:AdditionalItemProperty>
</cans:Item>
-<cans:Price>
  <cbns:PriceAmount currencyID="CAD">1.00</cbns:PriceAmount>
  <cbns:BaseQuantity unitCode="LTR">1</cbns:BaseQuantity>
</cans:Price>
If the Parties Choose to Use the Results
Ultimately, Who Rules? The Algorithm Manager or the Operations Manager?

TOWARDS A FRAMEWORK OF WEIGHTED OPTIONS

Who can/should/shall override whom?
Under what circumstances?
Based on what criteria?
How can this be ensured?

Rule source (de jure authority and/or de facto origins)
Rule subjectivity (degree of commitment)
Rule strength (gravity of non-compliance)
Ultimately, Who Rules?
The Algorithm Manager or the Operations Manager?

**Prioritize Operations Manager Agency**

**EMPOWERING STATEMENT**

To the extent the algorithm is not fulfilling a given requirement, the operations manager:

- can
- should
- shall

...override the algorithm manager and take control.

**Prioritize Algorithm Manager Agency**

**EMPOWERING STATEMENT**

To the extent the operations manager is not fulfilling a given requirement, the algorithm manager:

- can
- should
- shall

...override the operations manager and take control.
Ultimately, Who Rules? The Algorithm Manager or the Operations Manager?

**Prioritize Operations Manager Agency**

**CONSTRAINING STATEMENT**

To the extent the operations manager is fulfilling a given requirement, the algorithm manager:
- **cannot**
- **should not**
- **shall not**

...override the operations manager and take control.

**Prioritize Algorithm Manager Agency**

**CONSTRAINING STATEMENT**

To the extent the algorithm manager is fulfilling a given requirement, the operations manager:
- **cannot**
- **should not**
- **shall not**

...override the algorithm manager and take control.
Ultimately, Who Rules? The Algorithm Manager or the Operations Manager?

Prioritize Operations Manager Agency

DELEGATING STATEMENT

The operations manager may voluntarily delegate control to the algorithm manager: pro-actively upon request.

Prioritize Algorithm Manager Agency

DELEGATING STATEMENT

The algorithm manager may voluntarily delegate control to the operations manager: pro-actively upon request.
Ultimately, Who Rules? The Algorithm Manager or the Operations Manager?

POTENTIAL CRITERIA FOR INTERVENTION OR DELEGATION

- Better attainment
- Effectiveness
- Efficiency
- Sequence (in order to proceed)
- Information (in order to proceed)
- Priority of rules (defeasible logic)
- Mandate (modal logic)
Alpha Testing Now

Xalgorithms - Rule writing demo.

https://github.com/Xalgorithms
Video demo... https://tinyurl.com/yym4pgwd
Wrap-Up

Questions

General Framework:
• What conceptual map can illustrate RuleML vis-à-vis IoR functions?
• How might complementarity between IoR and RuleML be described?

Synergies:
• Can an IoR expedite the achievement of RuleML community goals?
• What challenges does RuleML face that an IoR may help to overcome?

Issues/Incongruities:
• Does any aspect of this overview seem incongruous with RuleML?
• Are there any apparent conceptual/technical/practical errors-omissions?

Experimentation:
• What exploratory projects might be useful over the next year?
• What challenges would the RuleML community put to IoR designers?