From Semantic Complex Event Processing to and Ubiquitous Pragmatic Web 4.0

Adrian Paschke
AG Corporate Semantic Web, Freie Universitaet Berlin
paschke@inf.fu-berlin
http://www.mi.fu-berlin.de/en/inf/groups/ag-csw/

Ontolog Summit 2015
Internet of Things: Toward Smart Networked Systems and Societies
Track C: Decision Making in Different Domains
12 February 2015
Agenda

• Introduction to Event Processing
• Semantic Complex Event Processing (SCEP)
• Event Processing Technical Society (EPTS)
  – Event Processing Standards Reference Model
  – Event Processing Reference Architecture
• Event Processing Function Patterns - Examples
  – Implementation Examples in the Prova Rule Engine (Platform Specific)
• Reaction RuleML Standard
  – Standardized Semantic Reaction Rules (Platform Independent)
• Vision - Ubiquitous Pragmatic Web 4.0
Agenda

• Introduction to Event Processing
• Semantic Complex Event Processing (SCEP)
• Event Processing Technical Society (EPTS)
  – Event Processing Standards Reference Model
  – Event Processing Reference Architecture
• Event Processing Function Patterns - Examples
  – Implementation Examples in the Prova Rule Engine (Platform Specific)
• Reaction RuleML Standard
  – Standardized Semantic Reaction Rules (Platform Independent)
• Vision - Ubiquitous Pragmatic Web 4.0
Event Processing vs. Databases

**Ex-Post Data Queries**

- Database Queries
  - Query Processor

**Real Time Data Processing**

- Event Subscriptions
  - Incoming Events
    - Processing moment of events
  - Time
    - Past
    - Future
Knowledge Value of Events

Value of Events

Proactive actions

Predictive Analysis

Complex event Processing

Real-Time

Late reaction or Long term report

Post-Processing

Historical Event

Before the event

At event

Some time after event
e.g. 1 hour

Time
Complex Events – What are they?

- Complex Events are aggregates, derivations, etc. of Simple Events

- Complex Event Processing (CEP) will enable, e.g.
  - Detection of state changes based on observations
  - Prediction of future states based on past behaviours
Complex Event Processing – What is it?

- **CEP** is about complex event detection and reaction to complex events
  - Efficient (near real-time) **processing** of large numbers of events
  - **Detection, prediction** and **exploitation** of relevant complex events
  - Supports **situation awareness**, track & trace, sense & respond
Complex Event Processing – What is it?

- Complex Event Processing (CEP) is a discipline that deals with event-driven behavior.
- Selection, aggregation, and event abstraction for generating higher level complex events of interest.
CEP – Why do we need it?

**Example Application Domains**

**BAM, ITSM**
Monitor and detect exceptional IT service and business behavior from occurred events.

**RTE**
Quick decisions, and reactions to threats and opportunities according to events in business transactions.

**Information Dissemination**
Valuable Information at the Right Time to the Right Recipient.

**Expert Decision Management**

**CEP Media**
Detect, Decide, Respond.
Core CEP Operations

1. Event Production

2. Event Definition

3. Event Selection

4. Event Aggregation

5. Event Handling

6. Event Consumption
The Many Roots of CEP…

Complex Event Processing (CEP) is a discipline that deals with event-driven behavior.
Example Event Processing Languages

- Reaction RuleML
- TIBCO
- XChangeEQ
- RuleCore
- AMiT
- Agent Logic
- Spade
- Streambase
- Oracle
- Apama
- Netcool Impact
- Esper
- Coral8
- EventZero
- Prova
- WBE

ECA / Reaction Rules
Agent Oriented
SQL Extension
State Oriented
Imperative/Script Based
Inference Rules

see DEBS 2009 EPTS Language Tutorial - http://www.slideshare.net/opher.etzion/debs2009-event-processing-languages-tutorial
Complex Events – How?

Example Event Algebra Operators:

- **Sequence Operator (•):** $(E1;E2)$
- **Disjunction Operator (\(\lor\)):** $(E1 \lor E2)$, at least one
- **Conjunction Operator (\(\land\)):** $(E1 \land E2)$
- **Simultaneous Operator (\(\equiv\)):** $(E1 = E2)$
- **Negation Operator (\(\neg\)):** $(E1 \land \neg E2)$
- **Quantification (Any):** Any(n) $E1$, when n events of type $E1$ occurs
- **Aperiodic Operator (Ap):** $Ap(E2, E1, E3)$, $E2$ Within $E1$ & $E3$
- **Periodic Operator (Per):** $Per(t, E1, E2)$, every $t$ time-steps in between $E1$ and $E2$
Example – Interval-based Sequence

Example: $D = A; (B; C)$
Event Detection Operators & Windowing

Time

Window \([t-5 \leftrightarrow t]\)

Processing time

Future

Events

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>t-8</td>
<td>t-7</td>
<td>t-6</td>
<td>t-5</td>
<td>t-4</td>
<td>t-3</td>
<td>t-2</td>
<td>t-1</td>
<td>t</td>
</tr>
</tbody>
</table>

Event Detection Pattern \([A; B]\)

Matches (dependent on event consumption policy)

Time \(t\)

\(t\) → 2 or 3

Time \(t-1\)

\(t-1\) → 1

Time \(t-2\)

\(t-2\) → 1

Time \(t-3\)

\(t-3\) → 2 or 3
Event Processing Agent and Event Processing Network

- Event Processing Network (EPN) is a collection of Event Processing Agents (EPA).

- The EPN describes the “programming in the large”, while each individual agent describes the “programming in the small”.

Agenda

- Introduction to Event Processing
- **Semantic Complex Event Processing (SCEP)**
- Event Processing Technical Society (EPTS)
  - Event Processing Standards Reference Model
  - Event Processing Reference Architecture
- Event Processing Function Patterns - Examples
  - Implementation Examples in the Prova Rule Engine *(Platform Specific)*
- Reaction RuleML Standard
  - Standardized Semantic Reaction Rules *(Platform Independent)*
- Vision - Ubiquitous Pragmatic Web 4.0
(Complex) Event Processing: events, complex events, event patterns, ...

+ Semantic technologies: rules & ontologies & structured linked data
Knowledge-based Event Processing

Event Stream

Knowledge Base
Example: Semantic CEP - Filter Pattern

Filter Pattern:
Stocks of companies, which have production facilities in Europe and produce products out of metal and have more than 10,000 employees.

Event Stream – stock quotes
{(Name, “OPEL”) (Price, 45)(Volume, 2000)(Time, 1)}
{(Name, “SAP”) (Price, 65)(Volume, 1000)(Time, 2)}

Semantic Knowledge Base
{(OPEL, is_a, car_manufacturer),
 (car_manufacturer, build, Cars),
 (Cars, are_build_from, Metall),
 (OPEL, hat_production_facilities_in, Germany),
 (Germany, is_in, Europe)}
{(OPEL, is_a, Major_corporation),
 (Major_corporation, have, over_10,000_employees)}
Summary Semantic CEP: Selected Benefits

- Event data becomes **declarative knowledge** while conforming to an underlying **formal semantics**
  - e.g., supports automated semantic enrichment and mediation between different heterogeneous domains and abstraction levels

- Reasoning over **situations and states** by event processing agents
  - e.g., *a process is executing when it has been started and not ended*
  - e.g., *a plane begins flying when it takes off and it is no longer flying after it lands*

- Better understanding of the **relationships between events**
  e.g., temporal, spatial, causal, .., relations between events, states, activities, processes
  - e.g., *a service is unavailable when the service response time is longer than X seconds and the service is not in maintenance state*
  - e.g., *a landing starts when a plane approaches. During landing mobile phones must be switched off*

- **Declarative knowledge-based processing** of events and reactions to situations
Agenda

• Introduction to Event Processing

• Semantic Complex Event Processing (SCEP)

• Event Processing Technical Society (EPTS)
  – Event Processing Standards Reference Model
  – Event Processing Reference Architecture

• Event Processing Function Patterns - Examples
  – Implementation Examples in the Prova Rule Engine (Platform Specific)

• Reaction RuleML Standard
  – Standardized Semantic Reaction Rules (Platform Independent)

• Vision - Ubiquitous Pragmatic Web 4.0
EPTS CEP /Reaction RuleML Standards Reference Model

Reference Architecture: Functional View

**Event Production**
- Publication, Retrieval

**Event Consumption**
- Dashboard, Apps, External Reaction

**Event Reaction**
- Assessment, Routing, Prediction, Discovery, Learning

**Complex Event Detection**
- Consolidation, Composition, Aggregation

**Event Analysis**
- Analytics, Transforms, Tracking, Scoring, Rating, Classification

**Event Preparation**
- Identification, Selection, Filtering, Monitoring, Enrichment

**Event Process Monitoring, Control**
- Event Correlations and Patterns
- Event Computations
- Event Selections
- Event Production/Consumption

**State Management**

**Event Actions**

**Process Updates**
- Resource Utilization
- High Availability
- Security
- Start/Stop

**Event and Complex Event**
- (Pattern, Control, Rule, Query, RegEx, etc)

**Design time**
- Event Definition, Modeling, (continuous) Improvement

**Run time**
- Event-Driven Reaction Rules

**Administration**
- Event and Complex Event Definition, Modeling, (continuous) Improvement
Agenda

• Introduction to Event Processing
• Semantic Complex Event Processing (SCEP)
• Event Processing Technical Society (EPTS)
  – Event Processing Standards Reference Model
  – Event Processing Reference Architecture
• Event Processing Function Patterns - Examples
  – Implementation Examples in the Prova Rule Engine (Platform Specific)
• Reaction RuleML Standard
  – Standardized Semantic Reaction Rules (Platform Independent)
• Vision - Ubiquitous Pragmatic Web 4.0
Event Processing Patterns

• Functions from Reference Architecture are a guide to possible event processing patterns

<table>
<thead>
<tr>
<th>Event Reaction</th>
<th>Assessment, Routing, Prediction, Discovery, Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex Event Detection</td>
<td>Consolidation, Composition, Aggregation</td>
</tr>
<tr>
<td>Event Analysis</td>
<td>Analytics, Transforms, Tracking, Scoring, Rating, Classification</td>
</tr>
<tr>
<td>Event Preparation</td>
<td>Identification, Selection, Filtering, Monitoring, Enrichment</td>
</tr>
</tbody>
</table>

see: Adrian Paschke, Paul Vincent, Alexandre Alves, Catherine Moxey: Tutorial on advanced design patterns in event processing. DEBS 2012: 324-334; www.slideshare.net/isvana/epts-debs2012-event-processing-reference-architecture-design-patterns-v204b
Example with SPARQL Query

% Filter for car manufacturer stocks and enrich the stock tick event with data from Wikipedia (DBPedia) about the manufacturer and the luxury cars

\[ \text{rcvMult}(\text{SID}, \text{stream}, “S&P500”, \text{inform}, \text{tick}(S,P,T)) :- \]
\[ \text{carManufacturer}(S, \text{Man}), % filter car manufacturers \]
\[ \text{luxuryCar}(\text{Man}, \text{Name}, \text{Car}), % query \]
\[ \text{EnrichedData} = [S, [\text{Man}, \text{Name}, \text{Car}]], % enrich with additional data \]
\[ \text{sendMsg}(\text{SID2}, \text{esb}, “epal”, \text{inform}, \text{happens}(\text{tick}(\text{EnrichedData}, P), T)) . \]

% rule implementing the query on DBPedia using SPARQL query

\[ \text{luxuryCar}(\text{Manufacturer}, \text{Name}, \text{Car}) :- \]
\[ \text{Query} = “\text{SELECT ?manufacturer ?name ?car % SPARQL RDF Query} \]
\[ \text{WHERE} \{ \text{?car } <\text{http://purl.org/dc/terms/subject}> <\text{http://dbpedia.org/resource/Category:Luxury_vehicles}> . \]
\[ \text{?car foaf:name } \text{name} . \]
\[ \text{?car dbo:manufacturer } \text{?man} . \]
\[ \text{?man foaf:name } \text{?manufacturer}. \]
\] ORDER by ?manufacturer ?name”,
\[ \text{sparql_select(Query, manufacturer(Manufacturer), name(Name), car(Car))}. \]
% stream1 is trusted but stream2 is not, so one solution is found: X=e1

@src(stream1) event(e1).
@src(stream2) event(e2).

%note, for simplicity this is just a simple fact, but more complicated rating, trust, reputation policies could be defined
trusted(stream1). %only event from ’’stream1’’ are trusted

ratedEvent(X):-
   @src(Source) %scoped reasoning on @src
   event(X) [trusted(Source)]. %guard on trusted sources

:-solve(ratedEvent(X)). % => X=e1 (but not e2)
% This reaction operates indefinitely. When the timer elapses (after 25 ms), the groupby map Counter is sent as part of the aggregation event and consumed in or group, and the timer is reset back to the second argument of @timer.

`groupby_rate() :-`

`Counter = ws.prova.eventing.MapCounter(), % Aggr. Obj. @group(g1) @timer(25,25,Counter) % timer every 25 ms rcvMsg(XID,stream,From,inform,tick(S,P,T)) % event [IM=T,Counter.incrementAt(IM)]. % aggr. operation`

`groupby_rate() :-`

% receive the aggregation counter in the or reaction `@or(g1) rcvMsg(XID,self,From,or,[Counter]), ...

... <consume the Counter aggretion object>.
rcvMsg(XID, esb, From, query-ref, buy(Product)) :-
    routeTo(Agent, Product), % derive processing agent
    % send order to Agent in new subconversation SID2
    sendMsg(SID2, esb, Agent, query-ref, order(From, Product)),
    % receive confirmation from Agent for Product order
    rcvMsg(SID2, esb, Agent, inform-ref, order(From, Product)).

% route to event processing agent 1 if Product is luxury
routeTo(epa1, Product) :- luxury(Product).
% route to epa 2 if Product is regular
routeTo(epa2, Product) :- regular(Product).

% a Product is luxury if the Product has a value over ...
luxury(Product) :- price(Product, Value), Value >= 10000.
% a Product is regular if the Product has a value below ...
regular(Product) :- price(Product, Value), Value < 10000.
Agenda

• Introduction to Event Processing
• Semantic Complex Event Processing (SCEP)
• Event Processing Technical Society (EPTS)
  – Event Processing Standards Reference Model
  – Event Processing Reference Architecture
• Event Processing Function Patterns - Examples
  – Implementation Examples in the Prova Rule Engine (Platform Specific)
• Reaction RuleML Standard
  – Standardized Semantic Reaction Rules (Platform Independent)
• Vision - Ubiquitous Pragmatic Web 4.0
The Reaction RuleML Family

see Ontolog Forum presentation: http://www.slideshare.net/swadpasc/reaction-rule-mladrianpaschke20140109long
see RuleML 2014 keynote: http://www.slideshare.net/swadpasc/paschke-rule-ml2014keynote
see SWAT4LS 2014 tutorial: http://www.slideshare.net/swadpasc/swat4-ls-2014tutorialrulesintro
Selected Reaction RuleML Algebra Operators

• Action Algebra
  Succession (Ordered Succession of Actions), Choice (Non-Deterministic Choice), Flow (Parallel Flow), Loop (Loops), Operator (generic Operator)

• Event Algebra
  Sequence (Ordered), Disjunction (Or), Xor (Mutual Exclusive), Conjunction (And), Concurrent, Not, Any, Aperiodic, Periodic, AtLeast, ATMost, Operator (generic Operator)

• Interval Algebra (Time/Spatio/Event/Action/… Intervals)
  During, Overlaps, Starts, Precedes, Meets, Equals, Finishes, Operator (generic Operator)

• Counting Algebra
  Counter, AtLeast, AtMost, Nth, Operator (generic Operator)

• Temporal operators
  Timer, Every, After, Any, Operator (generic Operator)

• Negation operators
  Naf, Neg, Negation (generic Operator)
Example - Typed Complex Event **Pattern** Definition

```xml
<Event key="#ce2" type="&ruleml;ComplexEvent">
  <signature> <!-- pattern signature definition -->
    <Sequence>
      <signature>
        <Event type="&ruleml;SimpleEvent">
          <signature><Event>...event_A...</Event></signature>
        </Event>
      </signature>
      <signature>
        <Event type="&ruleml;ComplexEvent" keyref="ce1"/>
      </signature>
    </Sequence>
    <signature>
      <Event type="cbe:CommonBaseEvent" iri="cbe.xml#xpointer(/CommonBaseEvent)"/>
    </signature>
    </Sequence>
  </signature>
</Event>

<Event key="#ce1">
  <signature> <!-- event pattern signature -->
    <Concurrent>
      <Event><meta><Time>...t3</Time></meta><signature>...event_B</signature></Event>
      <Event><meta><Time>...t3</Time></meta><signature>...event_C</signature></Event>
    </Concurrent>
    </signature>
</Event>

<Event key="#e1" keyref="#ce2"><arg>...</arg></Event>
```
Messaging Reaction Rules

```xml
<Rule>
  ...
  <do><Send><Message> ...query1 </Message></Send></do>
  <do><Send><Message> ...query2 </Message></Send></do>
  <on><Receive><Message> ...response2</Message> </Receive></on>
  <if> prove some conditions, e.g. make decisions on the received answers </if>
  <on><Receive><Message> ...response1 </Message></Receive></on>
  ....
</Rule>
```

Note: The „on“, „do“, „if“ parts can be in arbitrary combinations, e.g. to allow for a flexible workflow-style logic with subconversations and parallel branching logic.
Message Driven Routing
Event Routing in Event-Driven Workflows

rcvMsg(XID, Process, From, event, ["A"]) :-
    fork_b_c(XID, Process).

fork_b_c(XID, Process) :-
    @group(p1) rcvMsg(XID, Process, From, event, ["B"]),
    execute(Task1), sendMsg(XID, self, 0, event, ["D"]).

fork_b_c(XID, Process) :-
    @group(p1) rcvMsg(XID, Process, From, event, ["C"]),
    execute(Task2), sendMsg(XID, self, 0, event, ["E"]).

fork_b_c(XID, Process) :-
    % OR reaction group "p1" waits for either of the two
    event message handlers "B" or "C" and terminates the
    alternative reaction if one arrives
    @or(p1) rcvMsg(XID, Process, From, or, _).
Distributed Rule Base Interchange
Mobile Code

% Manager

upload_mobile_code(Remote,File) :
Writer = java.io.StringWriter(), % Opening a file fopen(File,Reader),
copy(Reader,Writer),
Text = Writer.toString(),
SB = StringBuffer(Text),
sendMsg(XID,esb,Remote,eval,consult(SB)).

% Service (Contractor)

rcvMsg(XID,esb,Sender,eval,[Predicate|Args]) :- derive([Predicate|Args]).
Rule Responder: http://responder.ruleml.org
Agenda

• Introduction to Event Processing
• Semantic Complex Event Processing (SCEP)
• Event Processing Technical Society (EPTS)
  – Event Processing Standards Reference Model
  – Event Processing Reference Architecture
• Event Processing Function Patterns - Examples
  – Implementation Examples in the Prova Rule Engine (Platform Specific)
• Reaction RuleML Standard
  – Standardized Semantic Reaction Rules (Platform Independent)
• Vision - Ubiquitous Pragmatic Web 4.0
The Pragmatic Web consists of the tools, practices and theories describing why and how people use information. In contrast to the Syntactic Web and Semantic Web the Pragmatic Web is not only about form or meaning of information, but about interaction which brings about e.g. understanding and commitments.

www.pragmaticweb.info
Example: Question-Answer Interaction
Syntax – Semantics - Pragmatics

• Syntax
  – “What time is it?” (Language)

• Semantics
  – Question about current time (Meaning)

• Pragmatics
  – An answer to the question is obligatory (even if time is unknown) (Understanding and Commitment)
Ubiquitous Pragmatic Web 4.0

- Pragmatic Web
- Semantic Web
- Syntactic Web

World Wide Web 1.0
- Connects Information

Desktop Computing
- Monolithic Systems Era

Desktop

Machine Understanding

Monolithic Systems Era

Desktop

Social Semantic Web 3.0,
Web of Services & Things,
Corporate Semantic Web
Connects People, Services and Things

Semantic Web 2.0
- Connects Knowledge

Smart Web TV

Massive Multi-player Web Gaming

Ubiquitous autonomic
Smart Services and Things

Ubiquitous Pragmatic Web 4.0
- Connects Intelligent Agents and Smart Things

Situation Aware Real-time Semantic
Complex Event Processing

Pragmatic Agent Ecosystems

W3C Open Web Platform

Ubiquitous Next Generation Agents and Social Connections
Questions?

Acknowledgement to the members of the Reaction RuleML technical group

Acknowledgment to the EPTS Reference Architecture working group members

Acknowledgement to the members of the Pragmantic Web community

Acknowledgement to the members of the Corporate Semantic Web group at FU Berlin
RuleML Online Community

• **RuleML MediaWiki** ([http://wiki.ruleml.org](http://wiki.ruleml.org))

• **Mailing lists** ([http://ruleml.org/mailman/listinfo](http://ruleml.org/mailman/listinfo))

  – Uncertainty Reasoning
  – Defeasible Logic
  – Reaction Rules
  – Multi-Agent Systems
  – …

• **RuleML sources are hosted on Github** ([https://github.com/RuleML](https://github.com/RuleML))
Further Reading – Surveys and Tutorials

  http://www.igi-global.com/book/handbook-research-emerging-rule-based/465

  http://link.springer.com/chapter/10.1007%2F978-3-642-04985-9_8

  http://link.springer.com/chapter/10.1007%2F978-3-642-24908-2_17


- Jon Riecke, Opher Etzion, François Bry, Michael Eckert, Adrian Paschke, Event Processing Languages, Tutorial at 3rd ACM International Conference on Distributed Event-Based Systems. July 6-9, 2009 - Nashville, TN
  http://www.slideshare.net/opher.etzion/debs2009-event-processing-languages-tutorial

  http://www.igi-global.com/book/handbook-research-emerging-rule-based/465
Further Reading – RuleML and Reaction RuleML

- Adrian Paschke: Reaction RuleML 1.0 for Rules, Events and Actions in Semantic Complex Event Processing, Proceedings of the 8th International Web Rule Symposium (RuleML 2014), Springer LNCS, Prague, Czech Republic, August, 18-20, 2014

  http://dx.doi.org/10.1007/978-3-642-16289-3_15

- Adrian Paschke, Harold Boley, Zhili Zhao, Kia Teymourian and Tara Athan: Reaction RuleML 1.0: Standardized Semantic Reaction Rules, 6th International Conference on Rules (RuleML 2012), Montpellier, France, August 27-31, 2012
  http://link.springer.com/chapter/10.1007%2F978-3-642-32689-9_9
  http://www.slideshare.net/swadpasc/reaction-ruleml-ruleml2012paschketutorial

  http://www.igi-global.com/chapter/rule-markup-languages-semantic-web/35852

  http://www.igi-global.com/book/handbook-research-emerging-rule-based/465

- Adrian Paschke and Harold Boley: Rule Responder: Rule-Based Agents for the Semantic-Pragmatic Web, in Special Issue on Intelligent Distributed Computing in International Journal on Artificial Intelligence Tools (IJAIT), Vol. 20,6, 2011
  https://www.researchgate.net/publication/220160498
Further Reading – EPTS Event Processing Reference Architecture and Event Processing Patterns

• Adrian Paschke, Paul Vincent, Alexandre Alves, Catherine Moxey: Advanced design patterns in event processing. DEBS 2012: 324-334; http://www.slideshare.net/isvana/epats-debs2012-event-processing-reference-architecture-design-patterns-v204b


Further Reading – Rule-Based Semantic CEP

- Corporate Semantic Web – Semantic Complex Event Processing
- Kia Teymourian, Adrian Paschke: Plan-Based Semantic Enrichment of Event Streams. ESWC 2014: 21-35
- Kia Teymourian, Adrian Paschke: Towards semantic event processing. DEBS 2009
Further Reading – Rules and Logic Programming, Prova


- **Prova Rule Engine** [http://www.prova.ws/](http://www.prova.ws/)

- **Prova 3 Semantic Web Branch**
    (The Java sources of the Prova 3 Semantic Web are managed on GitHub ([https://github.com/prova/prova/tree/prova3-sw](https://github.com/prova/prova/tree/prova3-sw))


- **Prova CEP examples**: [http://www.slideshare.net/isvana/epts-debs2012-event-processing-reference-architecture-design-patterns-v204b](http://www.slideshare.net/isvana/epts-debs2012-event-processing-reference-architecture-design-patterns-v204b)
Further Reading – Pragmatic Web and Corporate Semantic Web

• Pragmatic Web
  – [http://www.pragmaticweb.info](http://www.pragmaticweb.info)

• Corporate Semantic Web

• Presentations: [http://www.slideshare.net/swadpasc](http://www.slideshare.net/swadpasc)


• Schoop, M.; de Moor, A.; Dietz, J. (2006) "*The pragmatic web: a manifesto*", *CACM* 49 (5)