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

RuleML Realize your Knowledge

Freie Universität Berlin

CSW Corporate Semantic Web

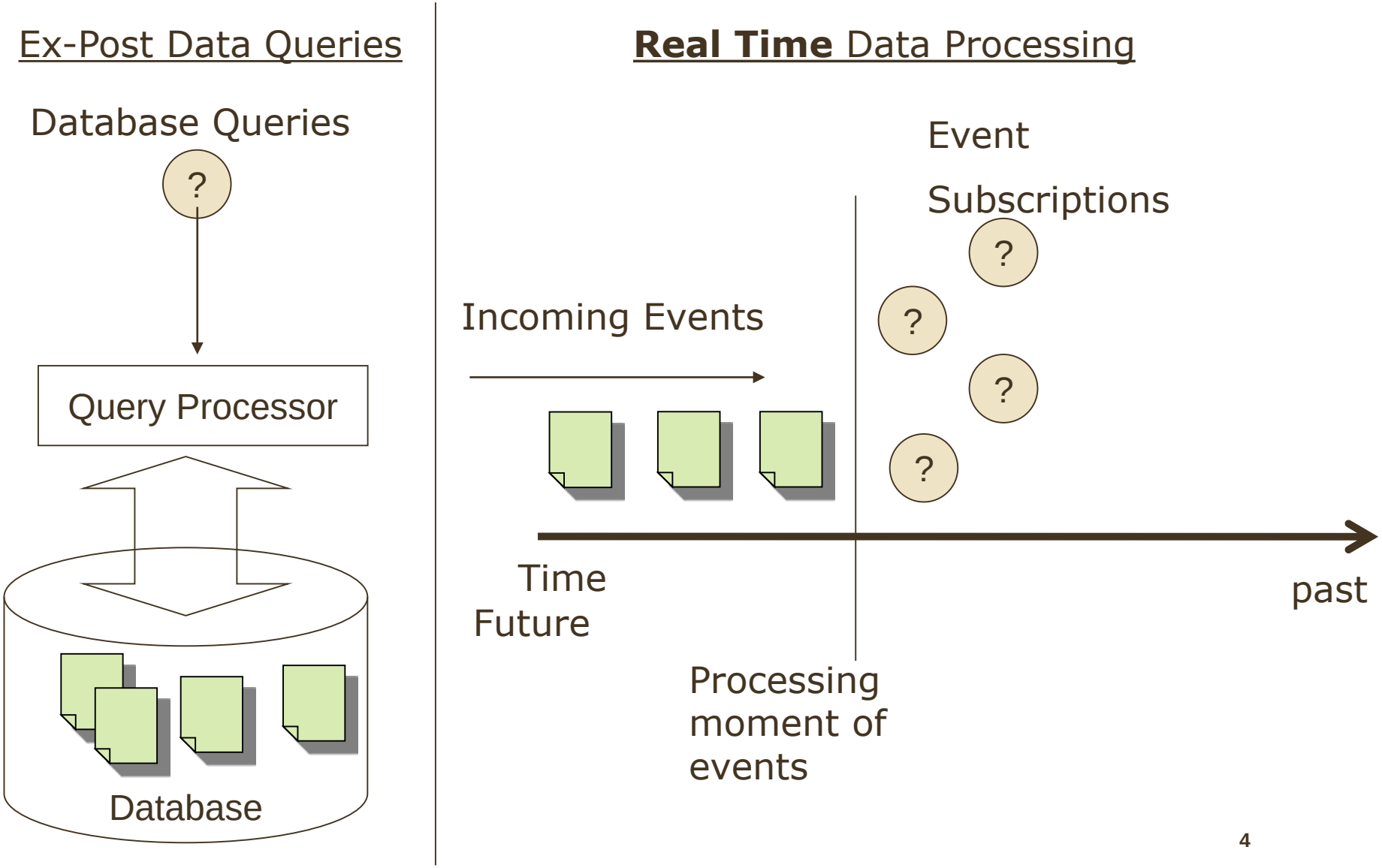
Agenda

- **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**

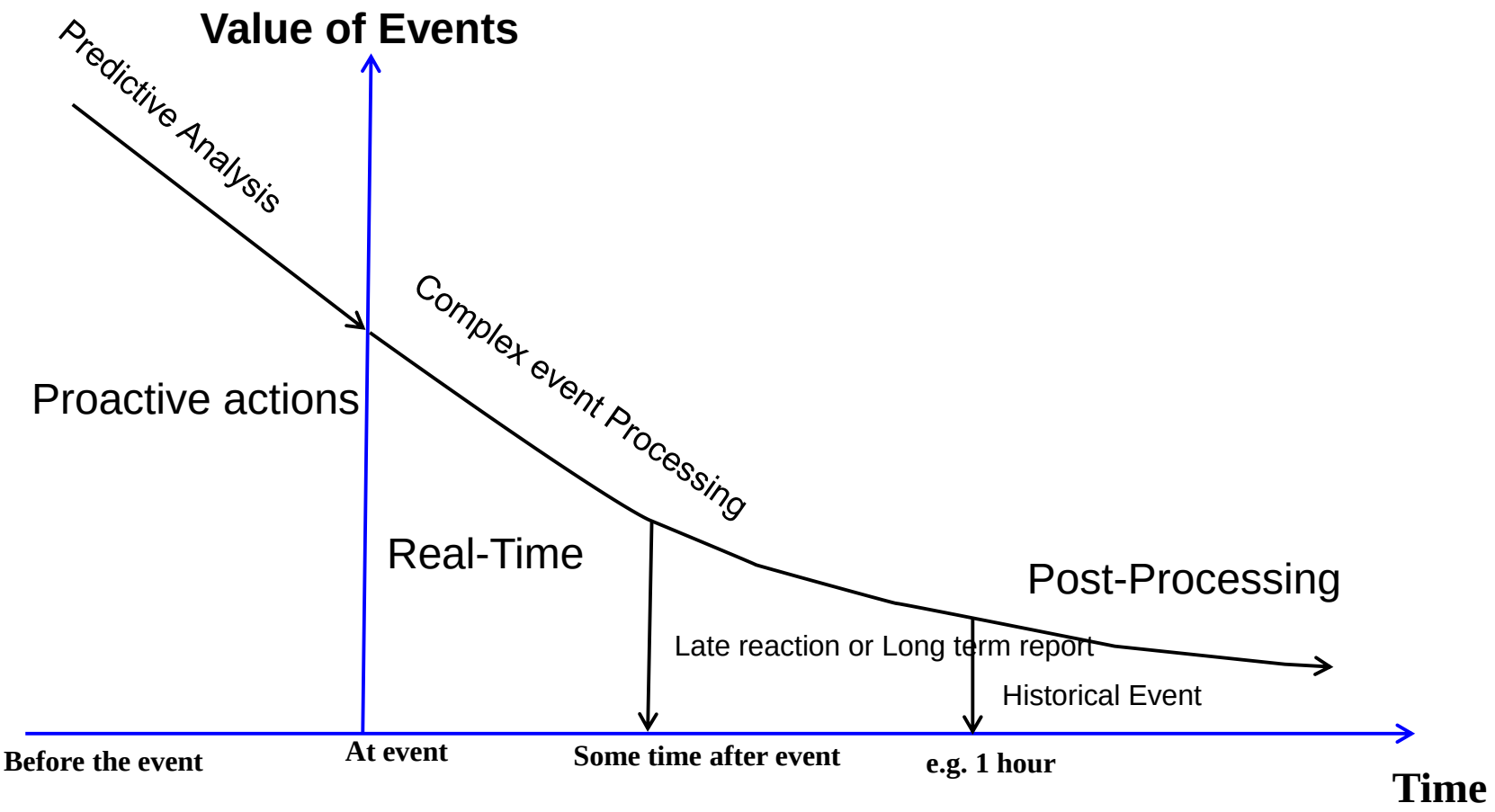
Agenda

- **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

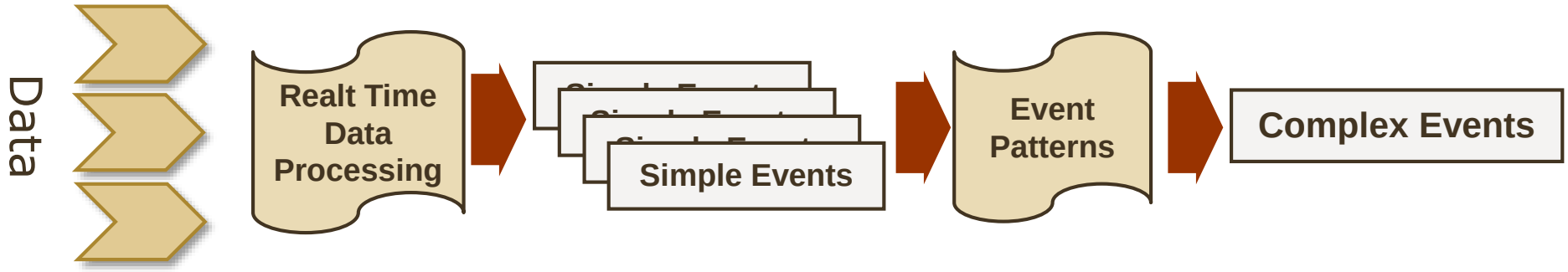


Knowledge Value of Events



Complex Events – What are they?

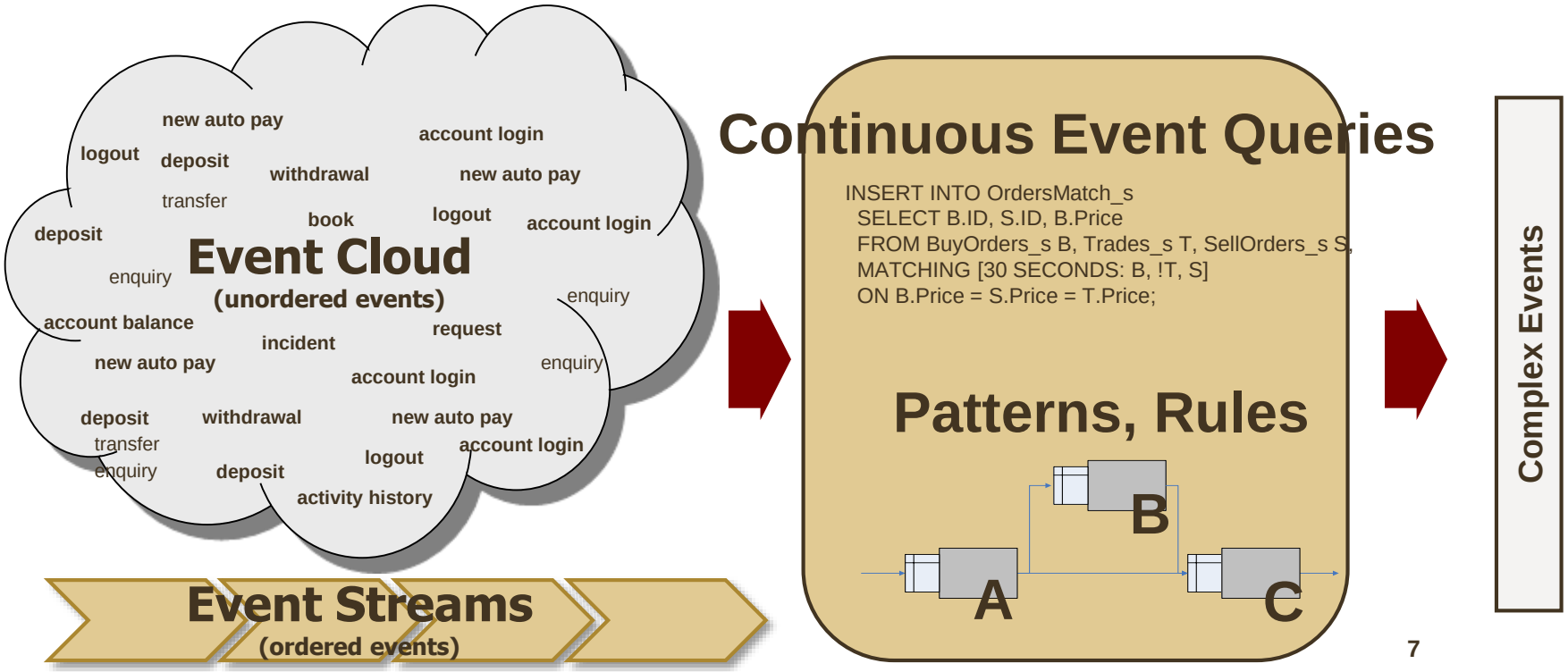
- **Complex Events are aggregates, derivations, etc. of Simple Events**



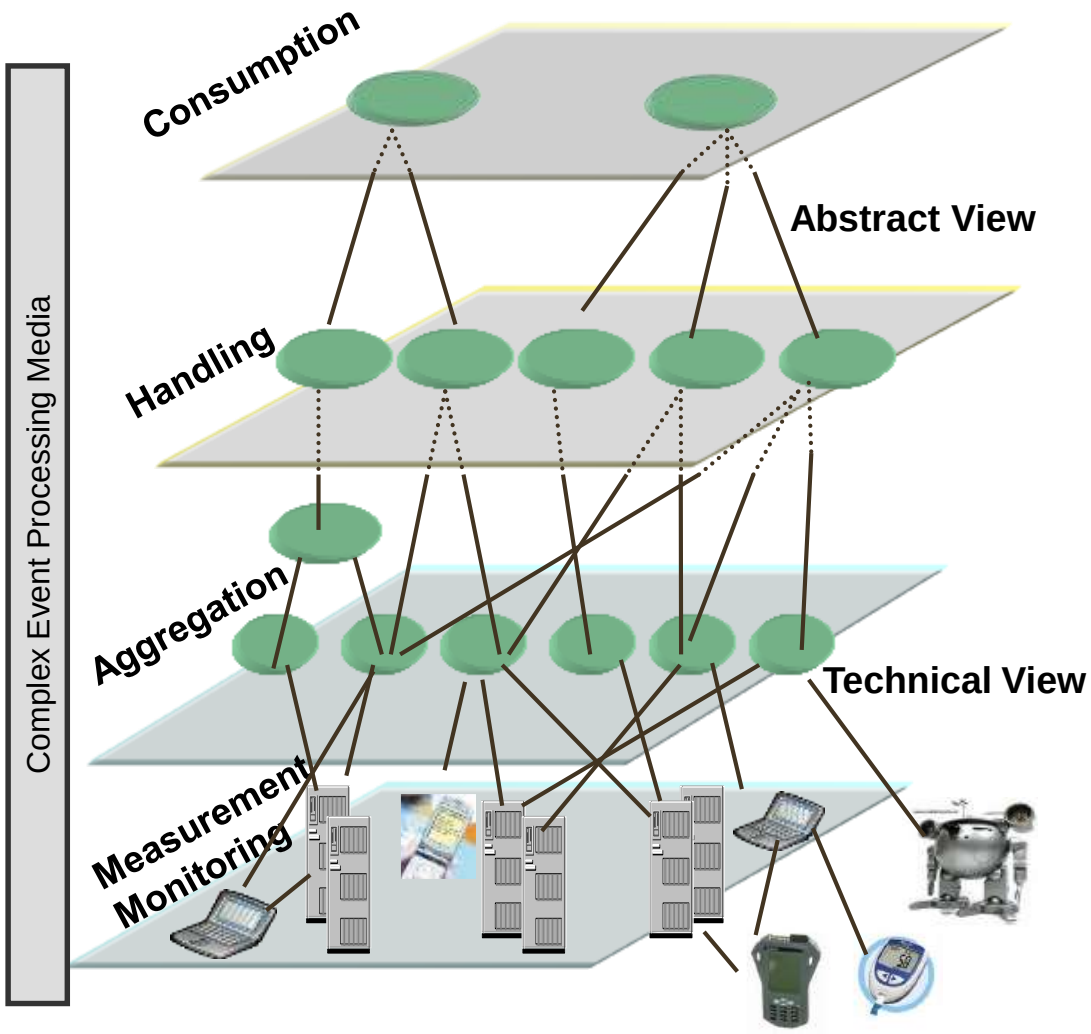
- Complex Event Processing (CEP) will enable, e.g.
  - **Detection** of state changes based on observations
  - **Prediction** of future states based on past behaviours

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?

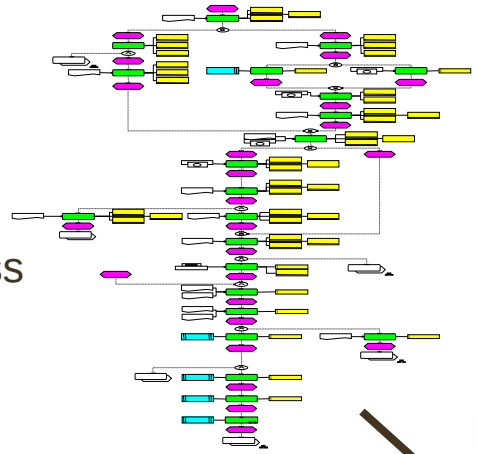


- Complex Event Processing (CEP) is a **discipline** that deals with event-driven behavior
- Selection, aggregation, and event abstraction for generating **higher level complex events of interest**

CEP – Why do we need it? Example Application Domains

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

RTE



BAM, ITSM

Monitor and detect exceptional IT service and business behavior from occurred events

Valuable Information at the Right Time to the Right Recipient

Information Dissemination

Expert Decision Management

Enterprise Decision Management

Expert Systems

Core CEP Operations

The Many Roots of CEP…

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

Active Databases

Agents

Distributed Event-based Computing

Event-driven Reaction Rules

Event-based Workflows / Business Processes

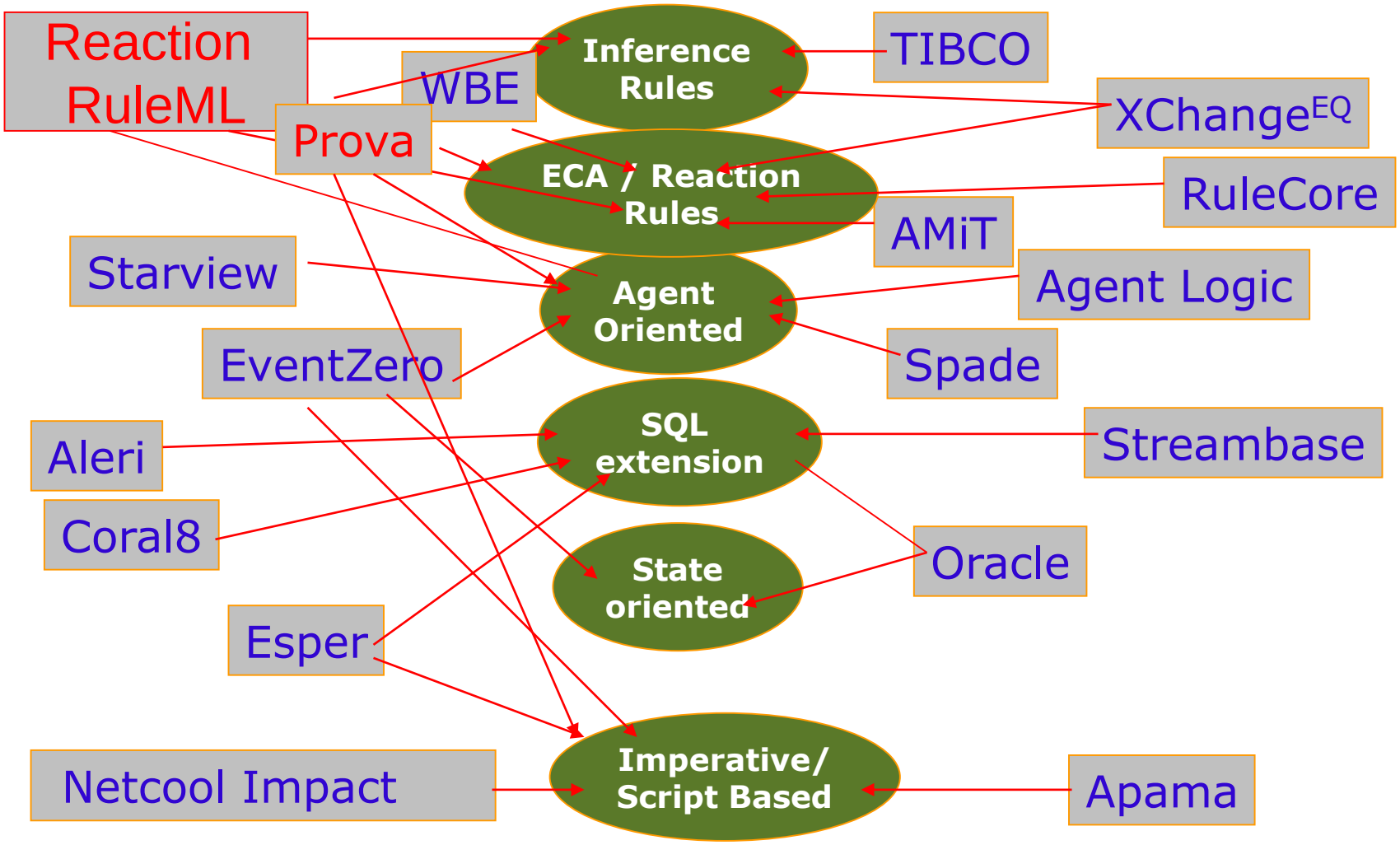
Discrete event simulation

High performance databases

Detect
Decide
Respond

Messaging / Middleware

Example Event Processing Languages



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

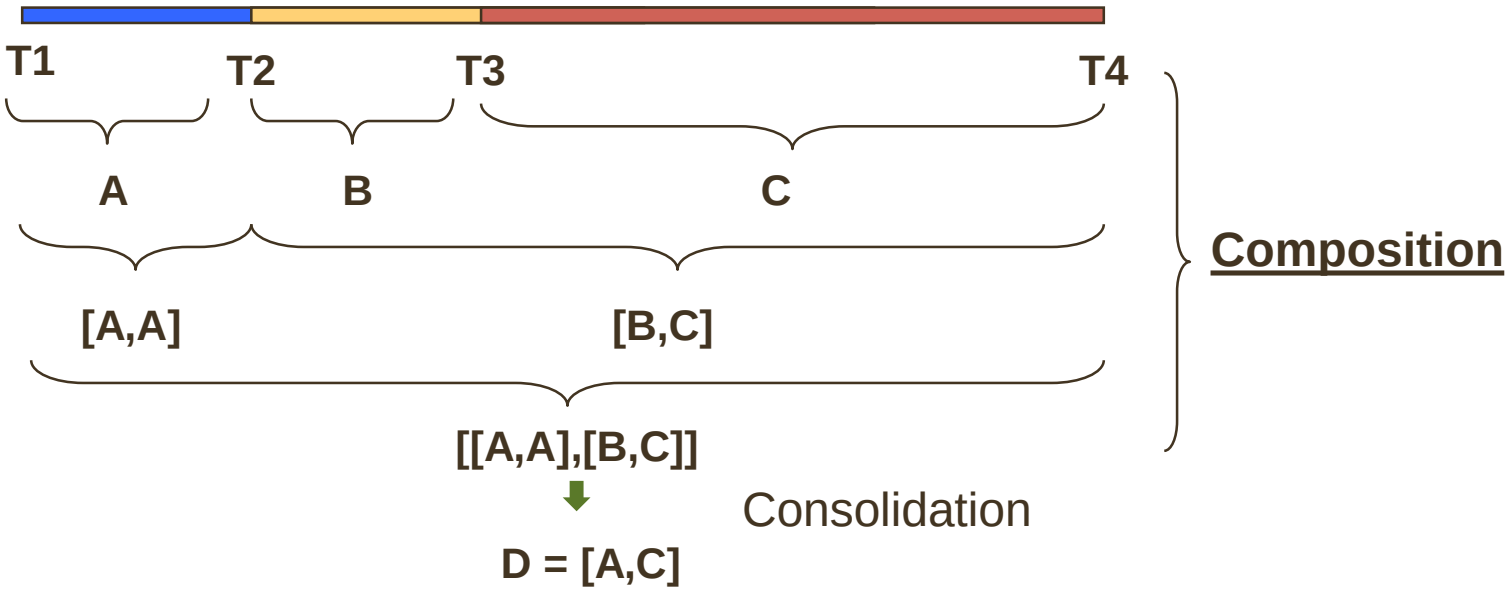
Complex Events – How?

Example Event Algebra Operators:

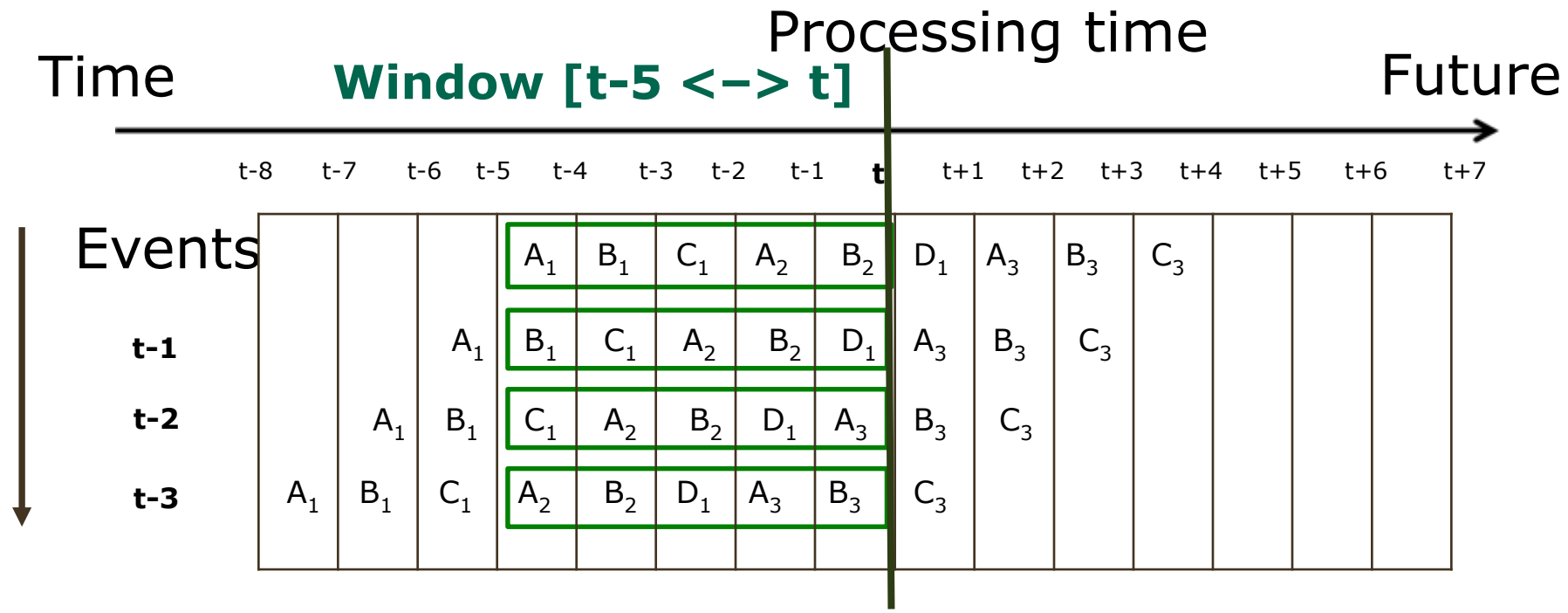
- **Sequence Operator (;):** *(E1;E2)*
- **Disjunction Operator (∨):** *(E1 ∨ E2)* , at least one
- **Conjunction Operator (∧):** (E1 ∧ E2)
- **Simultaneous Operator (=):** *(E1 = E2)*
- **Negation Operator (¬):** *(E1 ∧ ¬ E2)*
- **Quantification (Any):** Any(n) E1, when n events of type E1 occurs
- **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



Event Detection Pattern [A ; B]

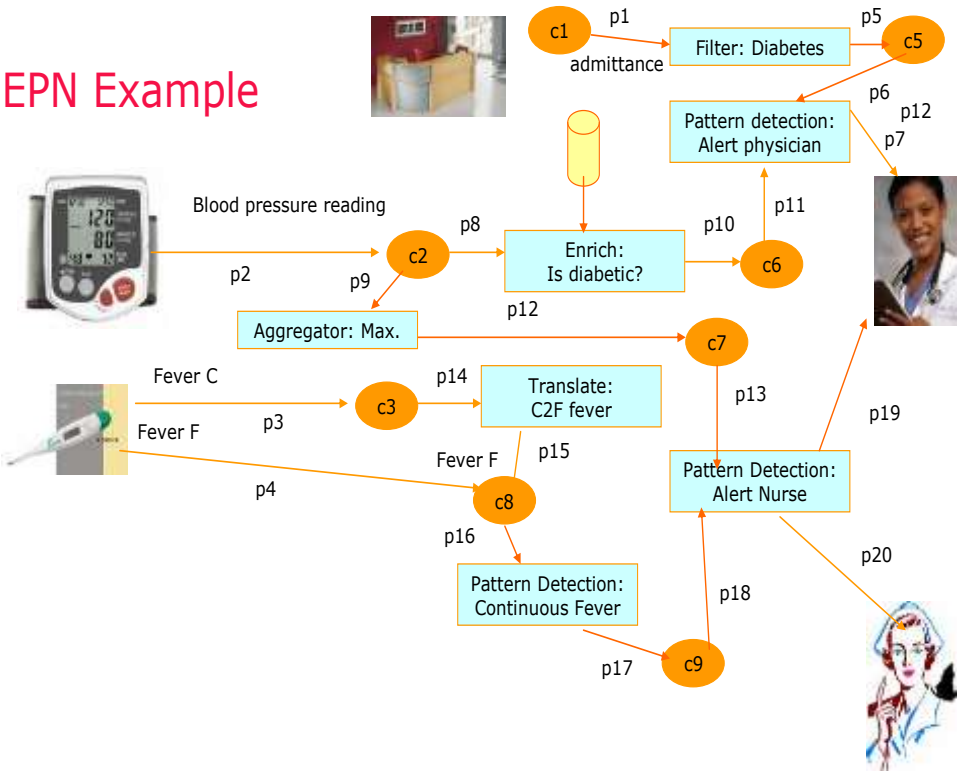
Matches (dependent on event consumption policy)

Time t → 2 or 3
Time t-1 → 1
Time t-2 → 1
Time 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".**



Opher Etzion, Peter Niblett: Event Processing in Action. Manning Publications Company 2010, ISBN 978-1-935182-21-4, pp. I-XXIV, 1-360

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**

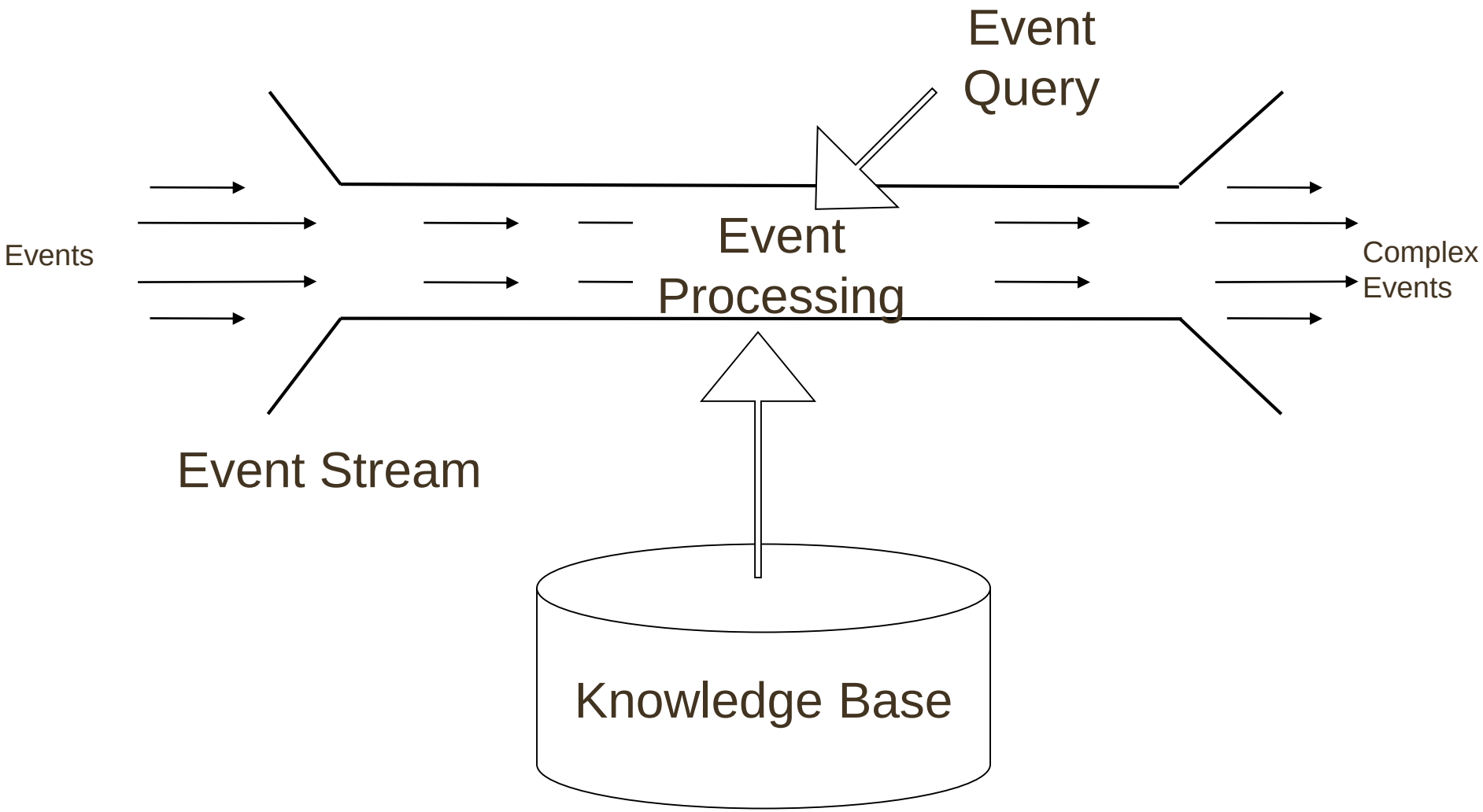
Semantic CEP: The Combination

(Complex) Event Processing: events, complex events, event patterns, …

+

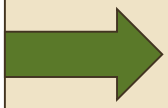
Semantic technologies: rules & ontologies & structured linked data

Knowledge-based Event Processing



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)}

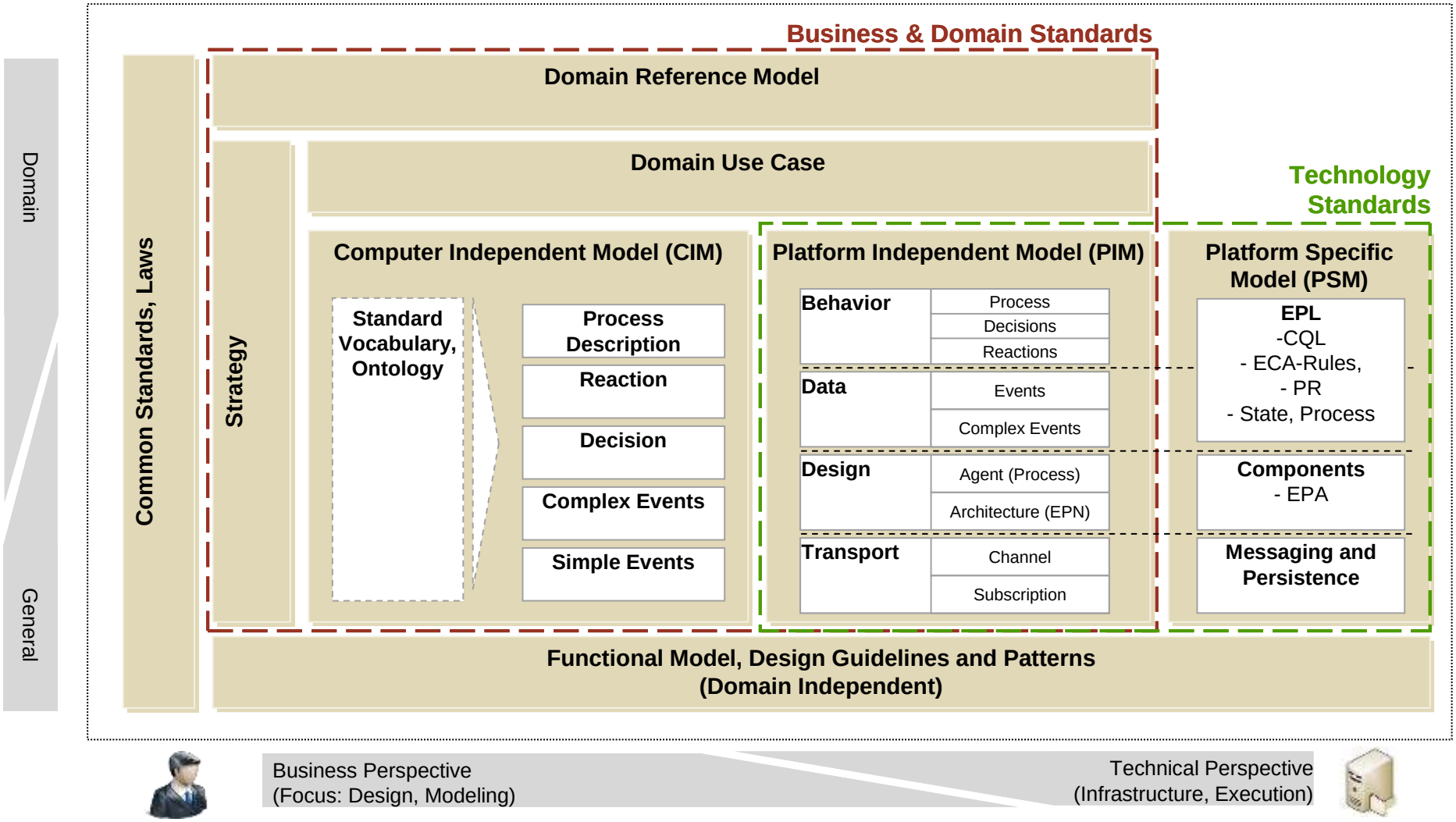
Summary Semantic CEP: Selected Benefits

- 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

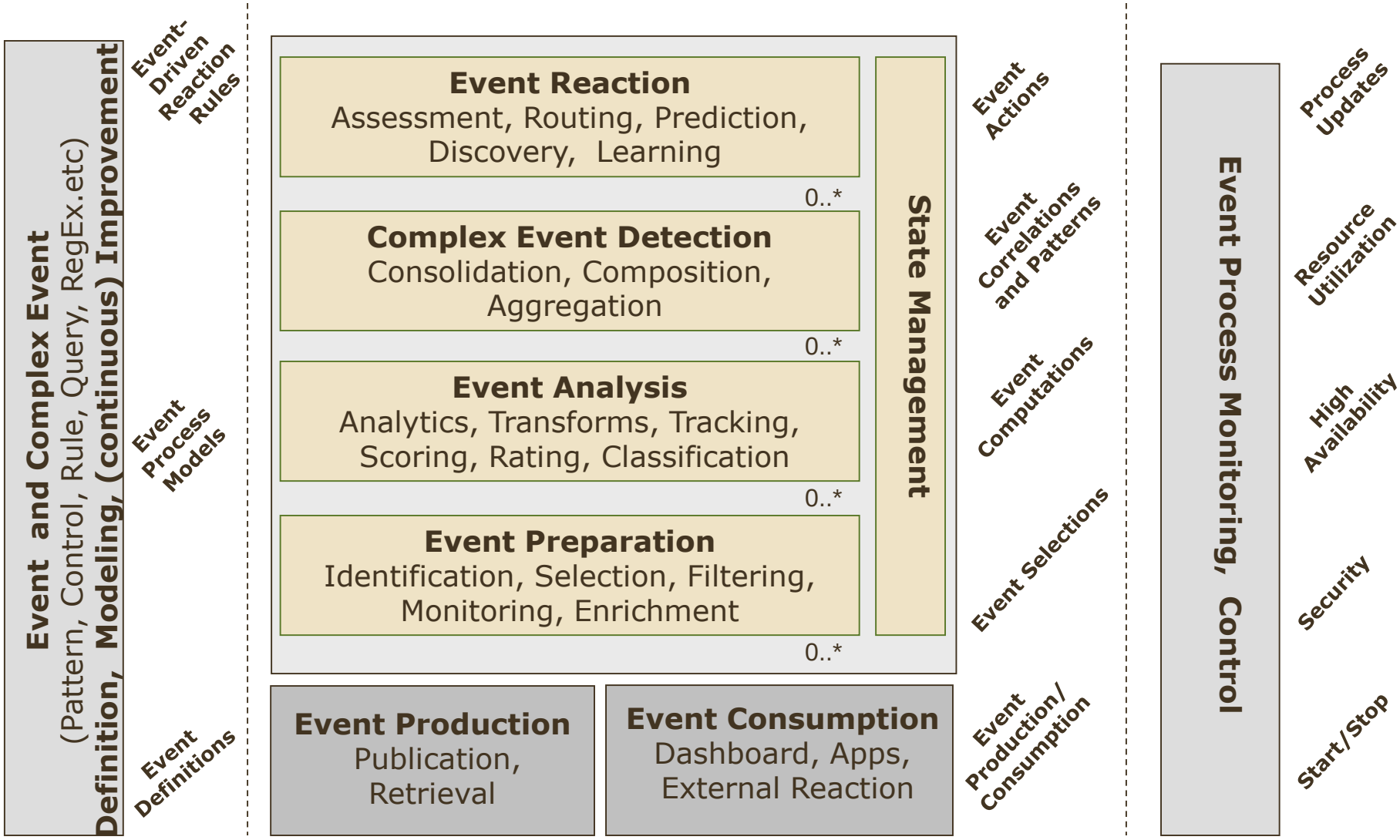
- **Introduction to Event Processing**
- **Semantic Complex Event Processing (SCEP)**
- **Event Processing Technical Society (EPTS)**
  - Event Processing Standards **Reference Model**
  - Event Processing **Reference Architecture**
- **Event Processing Function Patterns - Examples**
  - Implementation 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



see.: Adrian Paschke, Paul Vincent, Florian Springer: Standards for Complex Event Processing and Reaction Rules. RuleML America 2011: 128-139; http://www.slideshare.net/isvana/ruleml2011-cep-standards-reference-model

Reference Architecture: Functional View



Design time

Run time

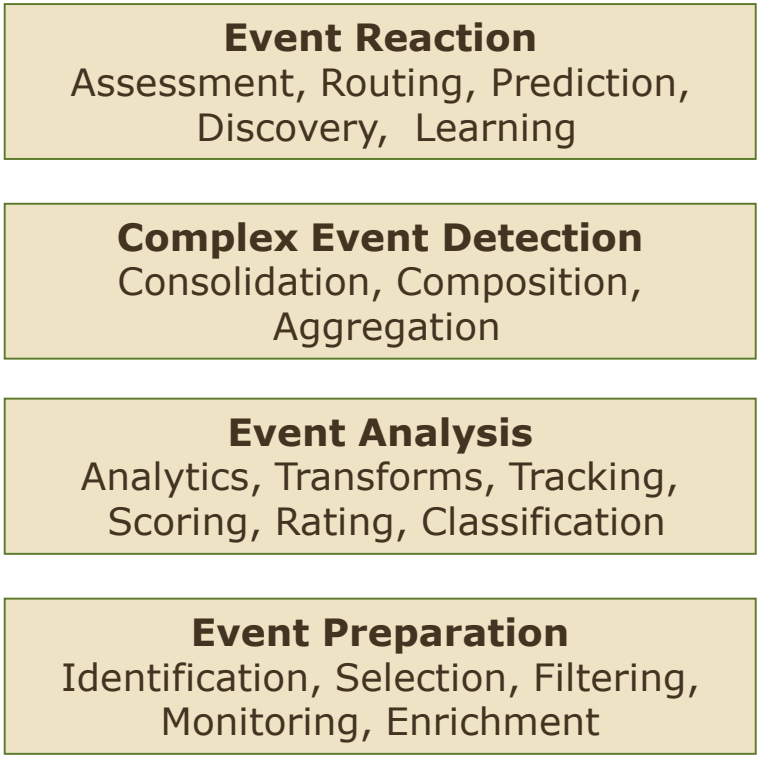
Administration

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**



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

Preparation:Enrichment:Implementations: Prova: 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
rcvMult(SID,stream,"S&P500", inform, tick(S,P,T)) :-
  carManufacturer(S,Man),    % filter car manufacturers
  luxuryCar(Man,Name,Car), % query
  EnrichedData = [S,[Man,Name,Car]], % enrich with additional data
  sendMsg(SID2,esb,"epa1", inform, happens(tick(EnrichedData,P),T).

% rule implementing the query on DBPedia using SPARQL query
luxuryCar(Manufacturer,Name,Car) :-
  Query="SELECT ?manufacturer ?name ?car    % SPARQL RDF Query
        WHERE {?car <http://purl.org/dc/terms/subject>
        <http://dbpedia.org/resource/Category:Luxury_vehicles> .
                ?car foaf:name ?name .
                ?car dbo:manufacturer ?man .
                ?man foaf:name ?manufacturer.
        } ORDER by ?manufacturer ?name",
sparql_select(Query,manufacturer(Manufacturer),name(Name),car(Car)).
```

Analysis:Rating:Implementations: Prova: Metadata Scoped Reasoning and Guards

```
% 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)
```

Detection:Aggregation:Implementations: Prova: Example with Time Counter

```
  % 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 aggreation object>.
```

Reaction:Routing:Implementations:
Prova: Example with Agent (Sub-) Conversations

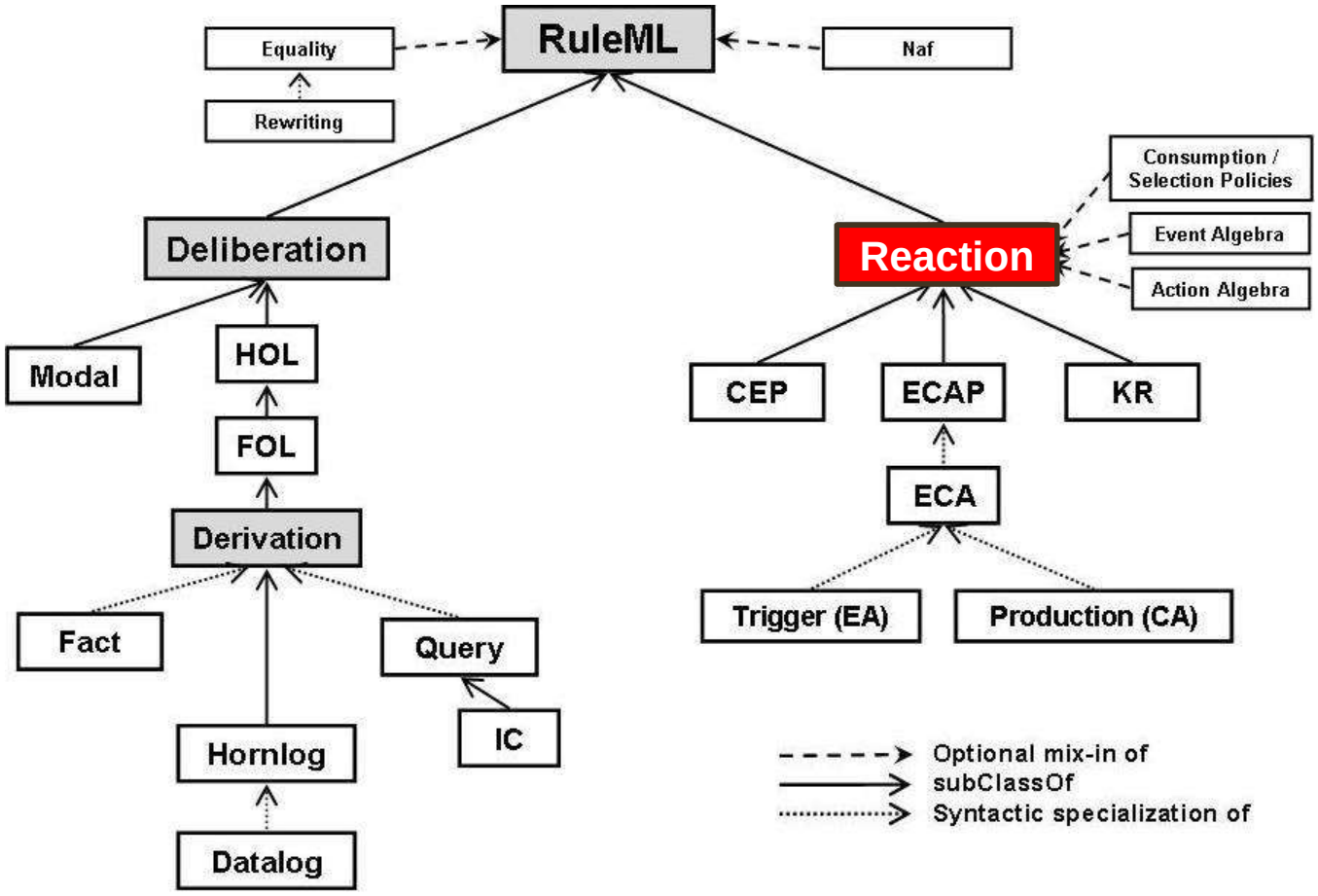
```
rcvMsg(XID,esb,From,query-ref,buy(Product) :-
 routeTo(Agent,Product), % derive processing agent
 % send order to Agent in new subconversation SID2
 sendMsg(SID2,esb,Agent,query-ref,order(From, Product)),
 % receive confirmation from Agent for Product order
 rcvMsg(SID2,esb,Agent,inform-ref,oder(From, 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 ha a value below …
regular(Product) :- price(Product,Value), Value < 10000.
```

corresponding serialization with Reaction RuleML <Send> and <Receive>

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-Determenistic Choice),** ***Flow*** **(Parallel Flow),** ***Loop*** **(Loops),** ***Operator*** **(generic Operator)**

- **Event Algebra**
  Sequence **(Ordered),** ***Disjunction*** **(Or) ,** ***Xor*** **(Mutal 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*

```
<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>
      <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 Pattern defining **Event Templates** signature

```
<Event key="#e1" keyref="#ce2"><arg>…</arg></Event>
```

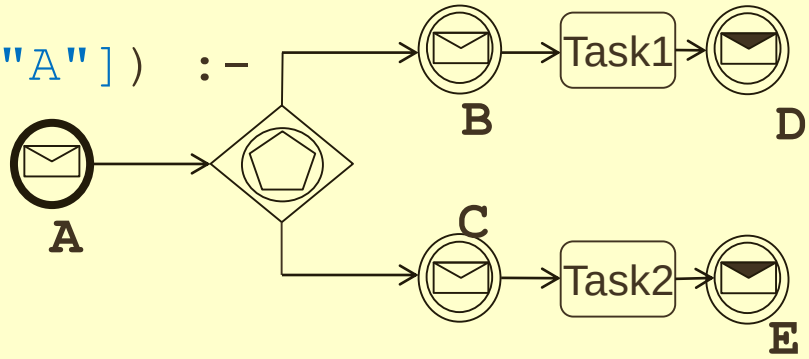
Event Instance

Messaging Reaction Rules

```
<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

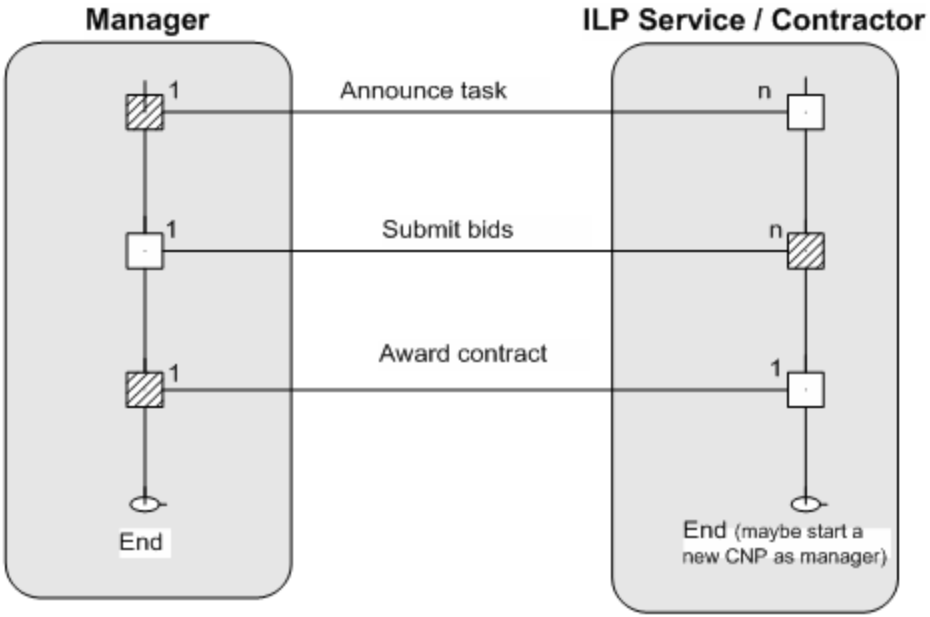
Message Driven Routing
Event Routing in Event-Driven Workflows



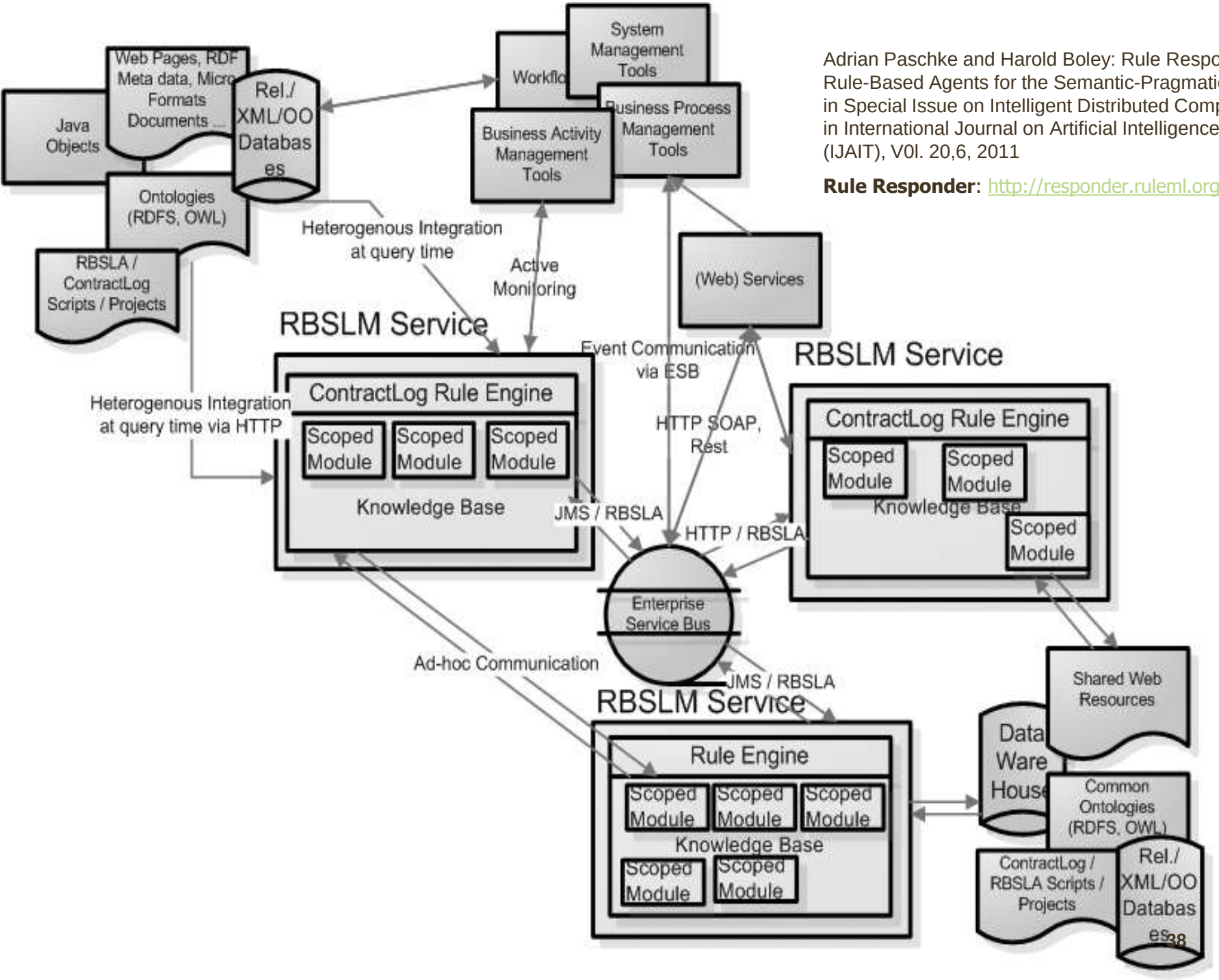

```
rcvMsg(XID,Process,From,event,["A"]) :-
  fork_b_c(XID, Process).


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]).
```



Contract Net Protocol



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), V0l. 20,6, 2011

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**

Pragmatic Web – Interaction and Commitment

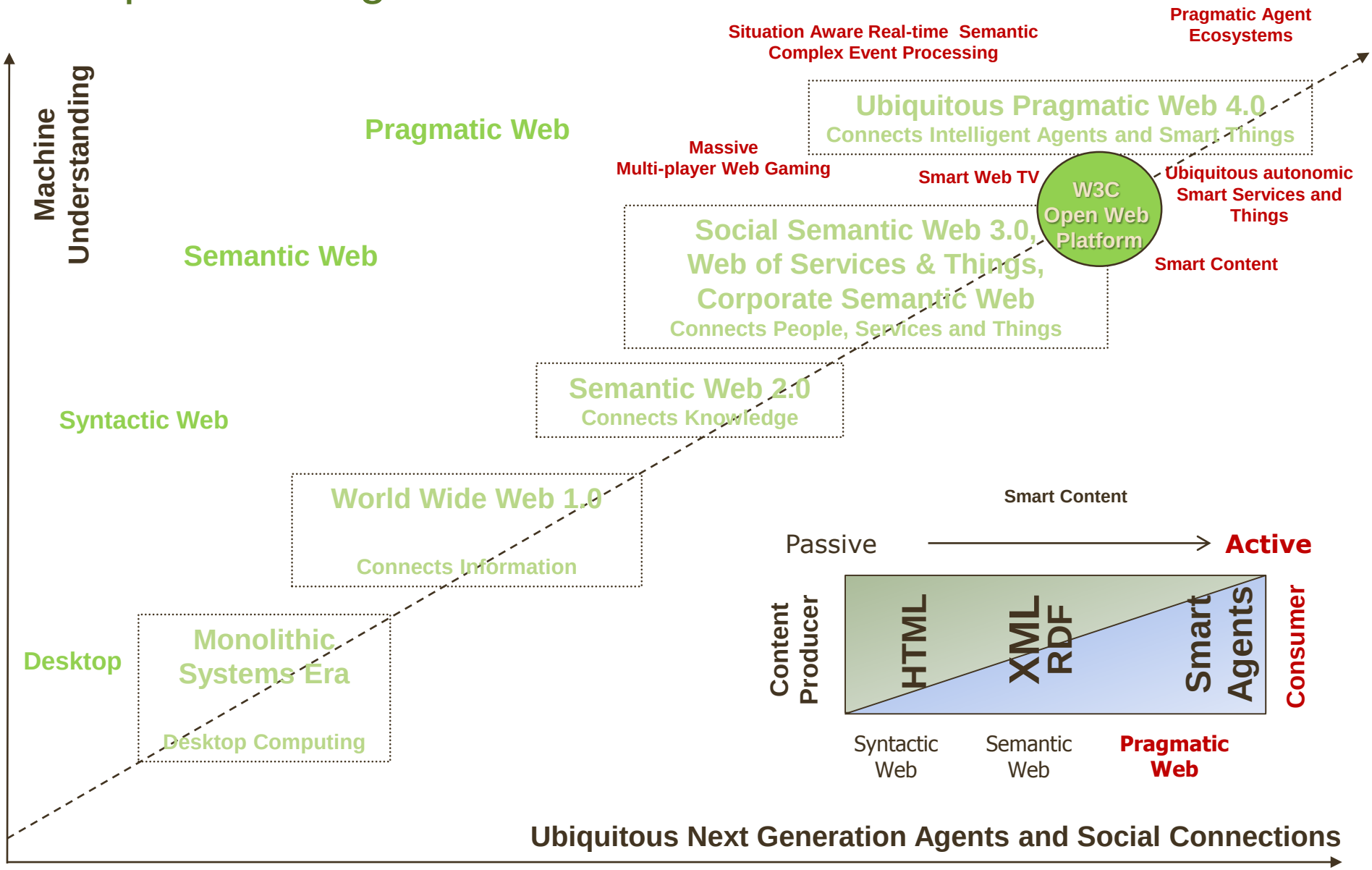
The Pragmatic Web consists of the tools, practices and theories describing why and how people use information. In contrast to the Syntactic Web and Semantic Web the Pragmatic Web is not only about form or meaning of information, but about interaction which brings about e.g. understanding and commitments.

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



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)**
- **Mailing lists (http://ruleml.org/mailman/listinfo)**
- **Technical Groups (http://wiki.ruleml.org/index.php/Organizational_Structure#Technical_Groups)**
  - Uncertainty Reasoning
  - Defeasible Logic
  - Reaction Rules
  - Multi-Agent Systems
  - …
- **RuleML sources are hosted on Github (https://github.com/RuleML)**

Further Reading – Surveys and Tutorials

- **Paschke, A., Boley, H.: Rules Capturing Event and Reactivity, in Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, IGI Publishing, ISBN:1-60566-402-2, 2009**
  http://www.igi-global.com/book/handbook-research-emerging-rule-based/465
  Sample Chapter: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=16435934&article=0&fd=pdf
- **Adrian Paschke, Alexander Kozlenkov: Rule-Based Event Processing and Reaction Rules. RuleML 2009: 53-66**
  http://link.springer.com/chapter/10.1007%2F978-3-642-04985-9_8
- **Adrian Paschke, Paul Vincent, Florian Springer: Standards for Complex Event Processing and Reaction Rules. RuleML America 2011: 128-139**
  http://link.springer.com/chapter/10.1007%2F978-3-642-24908-2_17
  http://www.slideshare.net/isvana/ruleml2011-cep-standards-reference-model
- **Adrian Paschke: The Reaction RuleML Classification of the Event / Action / State Processing and Reasoning Space. CoRR abs/cs/0611047 (2006)**
  http://arxiv.org/ftp/cs/papers/0611/0611047.pdf
- **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
- **Paschke, A., Boley, H.: Rule Markup Languages and Semantic Web Rule Languages, in Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, IGI Publishing, ISBN:1-60566-402-2, 2009**
  http://www.igi-global.com/chapter/rule-markup-languages-semantic-web/35852
  Sample: Chapter: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=18533385

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**
- **Harold Boley, Adrian Paschke, Omair Shafiq: RuleML 1.0: The Overarching Specification of Web Rules. RuleML 2010: 162-178**
  http://dx.doi.org/10.1007/978-3-642-16289-3_15
  http://www.cs.unb.ca/~boley/talks/RuleML-Overarching-Talk.pdf
- **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
- **Paschke, A., Boley, H.: Rule Markup Languages and Semantic Web Rule Languages, in Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, IGI Publishing, ISBN:1-60566-402-2, 2009**
  http://www.igi-global.com/chapter/rule-markup-languages-semantic-web/35852
  Sample chapter: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=18533385
- **Paschke, A., Boley, H.: Rules Capturing Event and Reactivity, in Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, IGI Publishing, ISBN:1-60566-402-2, 2009**
  http://www.igi-global.com/book/handbook-research-emerging-rule-based/465
  Sample Chapter: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=16435934&article=0&fd=pdf
- **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), V0l. 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/epts-debs2012-event-processing-reference-architecture-design-patterns-v204b
- **Adrian Paschke, Paul Vincent, Alexandre Alves, Catherine Moxey: Architectural and functional design patterns for event processing. DEBS 2011: 363-364** http://www.slideshare.net/isvana/epts-debs2011-event-processing-reference-architecture-and-patterns-tutorial-v1-2
- **Paschke, A., Vincent, P., and Catherine Moxey: Event Processing Architectures, *Fourth ACM International Conference on Distributed Event-Based Systems* (DEBS '10). ACM, Cambridge, UK, 2010.** http://www.slideshare.net/isvana/debs2010-tutorial-on-epts-reference-architecture-v11c
- **Paschke, A. and Vincent, P.: A reference architecture for Event Processing. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems* (DEBS '09). ACM, New York, NY, USA, 2009.**
- **Francois Bry, Michael Eckert, Opher Etzion, Adrian Paschke, Jon Ricke: Event Processing Language Tutorial, DEBS 2009, ACM, Nashville, 2009** http://www.slideshare.net/opher.etzion/debs2009-event-processing-languages-tutorial
- **Adrian Paschke. A Semantic Design Pattern Language for Complex Event Processing, AAAI Spring Symposium "Intelligent Event Processing", March 2009, Stanford, USA** http://www.aaai.org/Papers/Symposia/Spring/2009/SS-09-05/SS09-05-010.pdf
- **Paschke, A.: Design Patterns for Complex Event Processing, 2nd International Conference on Distributed Event-Based Systems (DEBS'08), Rome, Italy, 2008.** http://arxiv.org/abs/0806.1100v1

Further Reading – Rule-Based Semantic CEP

- **Corporate Semantic Web – Semantic Complex Event Processing**
  - http://www.corporate-semantic-web.de/semantic-complex-event-processing.html
- **Kia Teymourian, Adrian Paschke: Plan-Based Semantic Enrichment of Event Streams. ESWC 2014: 21-35**
- **Adrian Paschke: ECA-RuleML: An Approach combining ECA Rules with temporal interval-based KR Event/Action Logics and Transactional Update Logics CoRR abs/cs/0610167: (2006); http://arxiv.org/ftp/cs/papers/0610/0610167.pdf**
- **Adrian Paschke: Semantic Rule-Based Complex Event Processing. RuleML 2009: 82-92 http://link.springer.com/chapter/10.1007%2F978-3-642-04985-9_10**
- **Adrian Paschke, Alexander Kozlenkov, Harold Boley: A Homogeneous Reaction Rule Language for Complex Event Processing, In Proc. 2nd International Workshop on Event Drive Architecture and Event Processing Systems (EDA-PS 2007) at VLDB 2007; http://arxiv.org/pdf/1008.0823v1.pdf**
- **Adrian Paschke: ECA-LP / ECA-RuleML: A Homogeneous Event-Condition-Action Logic Programming Language CoRR abs/cs/0609143: (2006); http://arxiv.org/ftp/cs/papers/0609/0609143.pdf**
- **Teymourian, K., Coskun, G., and Paschke, A. 2010. Modular Upper-Level Ontologies for Semantic Complex Event Processing. In Proceeding of the 2010 Conference on Modular ontologies: Proceedings of the Fourth international Workshop (WoMO 2010) O. Kutz, J. Hois, J. Bao, and B. Cuenca Grau, Eds. Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, The Netherlands, 81-93. http://dl.acm.org/citation.cfm?id=1804769**
- **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), V0l. 20,6, 2011 https://www.researchgate.net/publication/220160498**
- **Kia Teymourian, Adrian Paschke: Towards semantic event processing. DEBS 2009**

Further Reading –Rules and Logic Programming, Prova

- **Adrian Paschke: Rules and Logic Programming for the Web.** A. Polleres et al. (Eds.): Reasoning Web 2011, LNCS 6848, pp. 326–381, 2011. http://dx.doi.org/10.1007/978-3-642-23032-5_6
  http://rw2011.deri.ie/lectures/05Paschke_Rules_and%20Logic_Programming_for_the_Web.pdf
- **Prova Rule Engine http://www.prova.ws/**
  - Prova 3 documentation http://www.prova.ws/index.html?page=documentation.php
    - Journal: 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), V0l. 20,6, 2011
  - **Prova 3 Semantic Web Branch**
    - Paper: Paschke, A.: A Typed Hybrid Description Logic Programming Language with Polymorphic Order-Sorted DL-Typed Unification for Semantic Web Type Systems, OWL-2006 (OWLED'06), Athens, Georgia, USA.
    - pre-compiled Prova 3 version with Semantic Web support from http://www.csw.inf.fu-berlin.de/teaching/ws1213/prova-sw.zip.
      (The Java sources of the Prova 3 Semantic Web are managed on GitHub (https://github.com/prova/prova/tree/prova3-sw)
  - Prova ContractLog KR: http://www.rbsla.ruleml.org/ContractLog.html
    - Paschke, A.: Rule-Based Service Level Agreements - Knowledge Representation for Automated e-Contract, SLA and Policy Management, Book ISBN 978-3-88793-221-3, Idea Verlag GmbH, Munich, Germany, 2007.
      http://rbsla.ruleml.org
  - Prova CEP examples: 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
- **Corporate Semantic Web**
  - http://www.corporate-semantic-web.de/

- **Presentations:** http://www.slideshare.net/swadpasc

- **Hans Weigand, Adrian Paschke: The Pragmatic Web: Putting Rules in Context. RuleML 2012: 182-192**

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

- **Adrian Paschke and Harold Boley. Rule Responder: Rule-based Agents for the Semantic-Pragmatic Web. International Journal on Artificial Intelligence Tools, 20(6):1043-1081, 2011, DOI: 10.1142/S0218213011000528**