Semantic Complex Event Processing

„The Future of Dynamic IT“

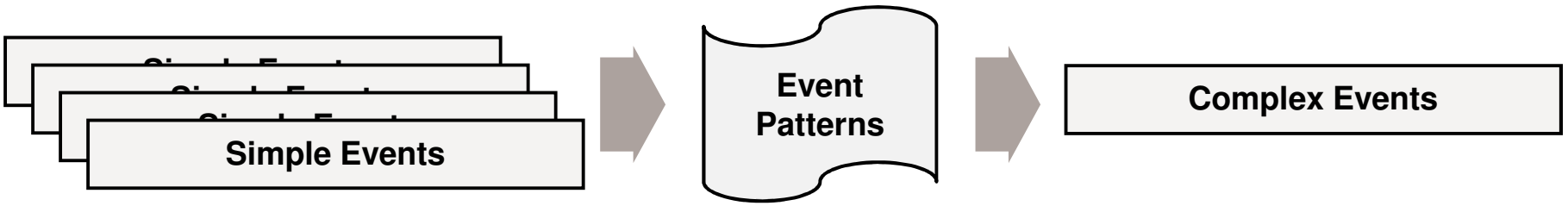
Paul Vincent
Adrian Paschke
Harold Boley

Agenda

- Complex Event Processing – What is it?
- Semantic Complex Event Processing (SCEP) – Why is semantics important in event processing?
- SCEP + Ontologies
- SCEP + (Reaction) Rules
  - Example: Reaction RuleML
- Use Cases for Semantics in CEP Systems
- Summary

Complex Events – What are they?

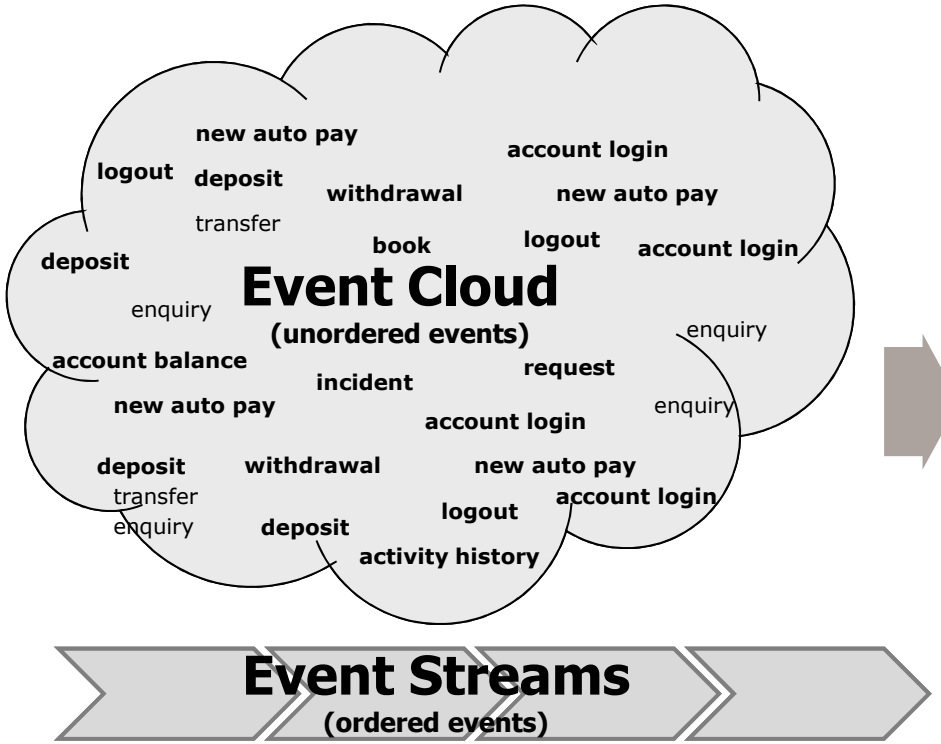
- **Complex Events** are aggregates, derivations, etc. of **Simple Events**

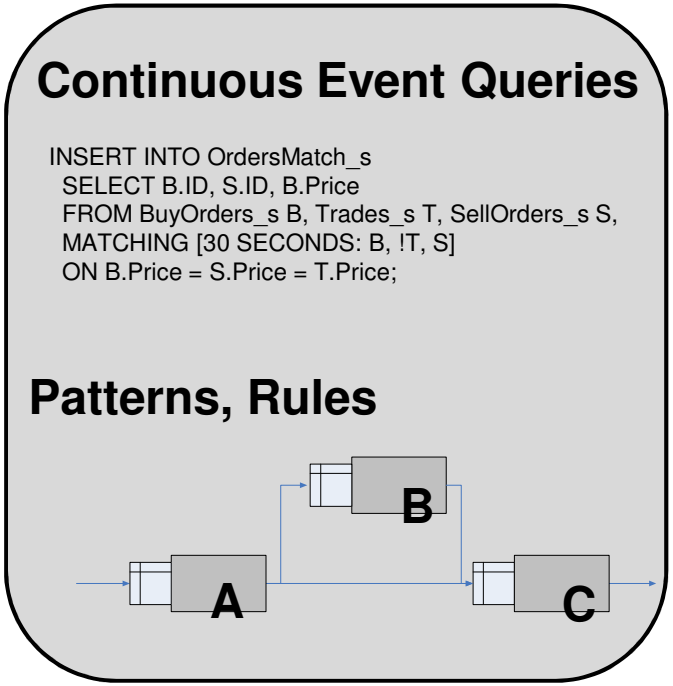


- Complex Event Processing (CEP) will enable, e.g.
  - **Detection** of state changes based on observations
  - **Prediction** of future states based on past behaviors

Complex Event Processing – What is it?

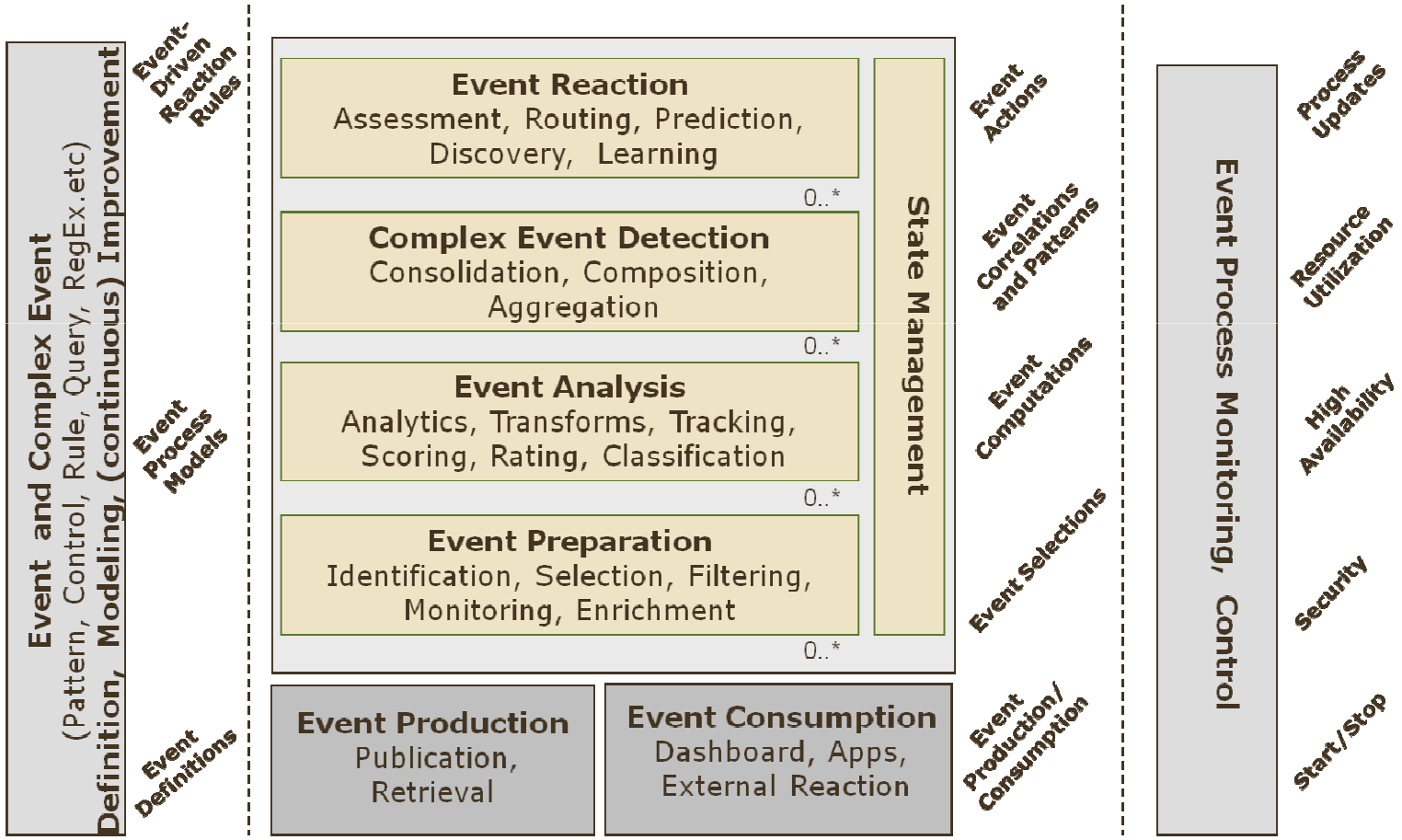
- CEP is about complex event detection and reaction to complex events
  - Efficient (near real-time) **processing** of large numbers of events
  - **Detection**, **prediction** and **exploitation** of relevant complex events
  - Situation awareness, **track & trace**, **sense & respond**





Complex Events

CEP Functions

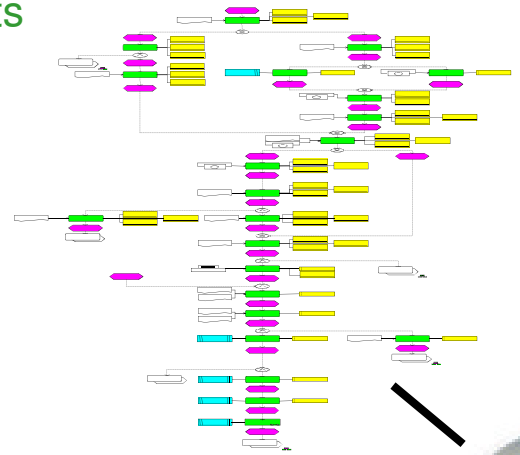


Source: Event Processing Technical Society Reference Architecture Working Group

Some CEP Scenarios …

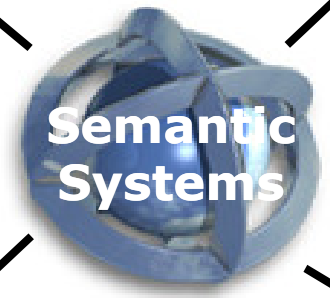
Real Time Enterprise

Reactions to threats and opportunities according to events in business transactions



BAM, Agile ITSM and BPM

Monitor and detect IT service, SLA, and business behavior exceptions from observed events



Detect
Decide
Respond

Valuable Information at the Right Time to the Right Recipient

Dynamic Information Dissemination

Real Time Decision Management

Enterprise Decision Management

Real-time Decision Guidance and Expert Decision Support

The Many Roots of Event Processing

Agents

Active Databases

Distributed Event-based Computing

Event-driven Rules

Event-based Workflows / Business Processes

Discrete event simulation

High performance databases

Detect
Decide
Respond

Messaging / Middleware

Semantic CEP: The Combination

Semantic CEP (or SCEP) combines approaches of

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

  +

- Semantic technologies: ontologies, definitions + behavior rules

Semantic CEP: The Benefits

- Event data becomes **meaningful information** / **declarative knowledge** while conforming to an underlying **formal semantics**
  - e.g., automated mediation between different heterogeneous domains and abstraction levels
- Better understanding of **situations (states)** by machines (agents)
  - e.g., a process is executing when it has been started and not ended
- 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
- Declarative **processing of events** and **reaction to situations**
  - Semantically grounded event-driven rules (= **reaction rules**)

Ontologies used for SCEP

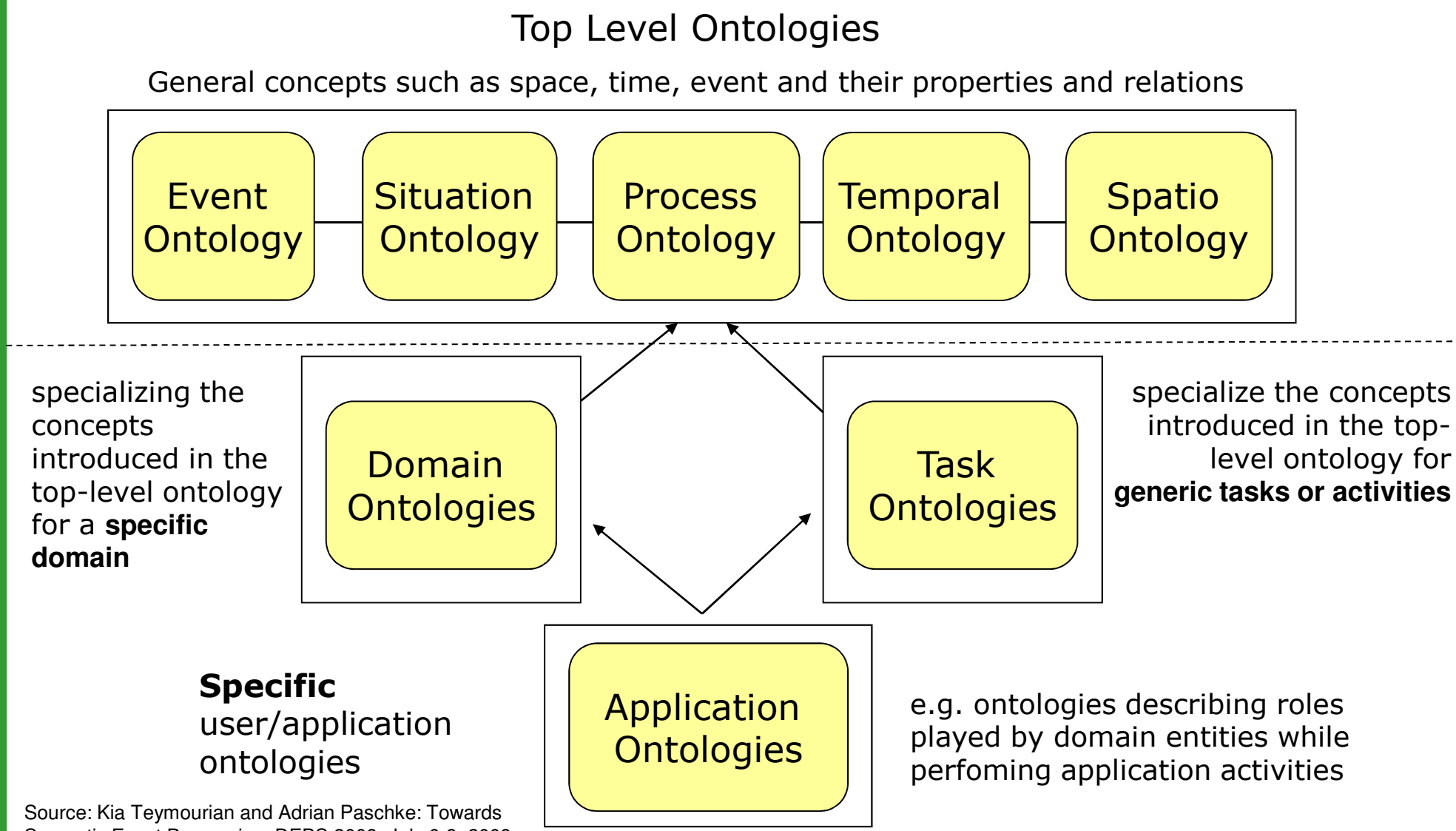
- **Top-Level Ontologies required for SCEP**
  (core selection)
  - **Spatial**
  - **Temporal**
  - **Event**
  - **Situation**
  - **Process (can be further specialized by domain ontologies such as OWL-S, WSMO, PSL)**
  - **Actor/Agent (can be further specialized, by special ontologies such as FIPA, ACL, ...)**
  - **Action: (can be used in e.g. RIF, RuleML, …)**

- **Domain Ontologies for application verticals**
  (samples domains of CEP applications)
  - **Healthcare - e.g., Hospital Activity Monitoring**
  - **Finance - e.g., Fraud Detection**
  - **Logistics and Cargo**
  - **Supply Chain Management**
  - **Insurance**
  - **Mortgage**

Examples of Ontologies which include Events

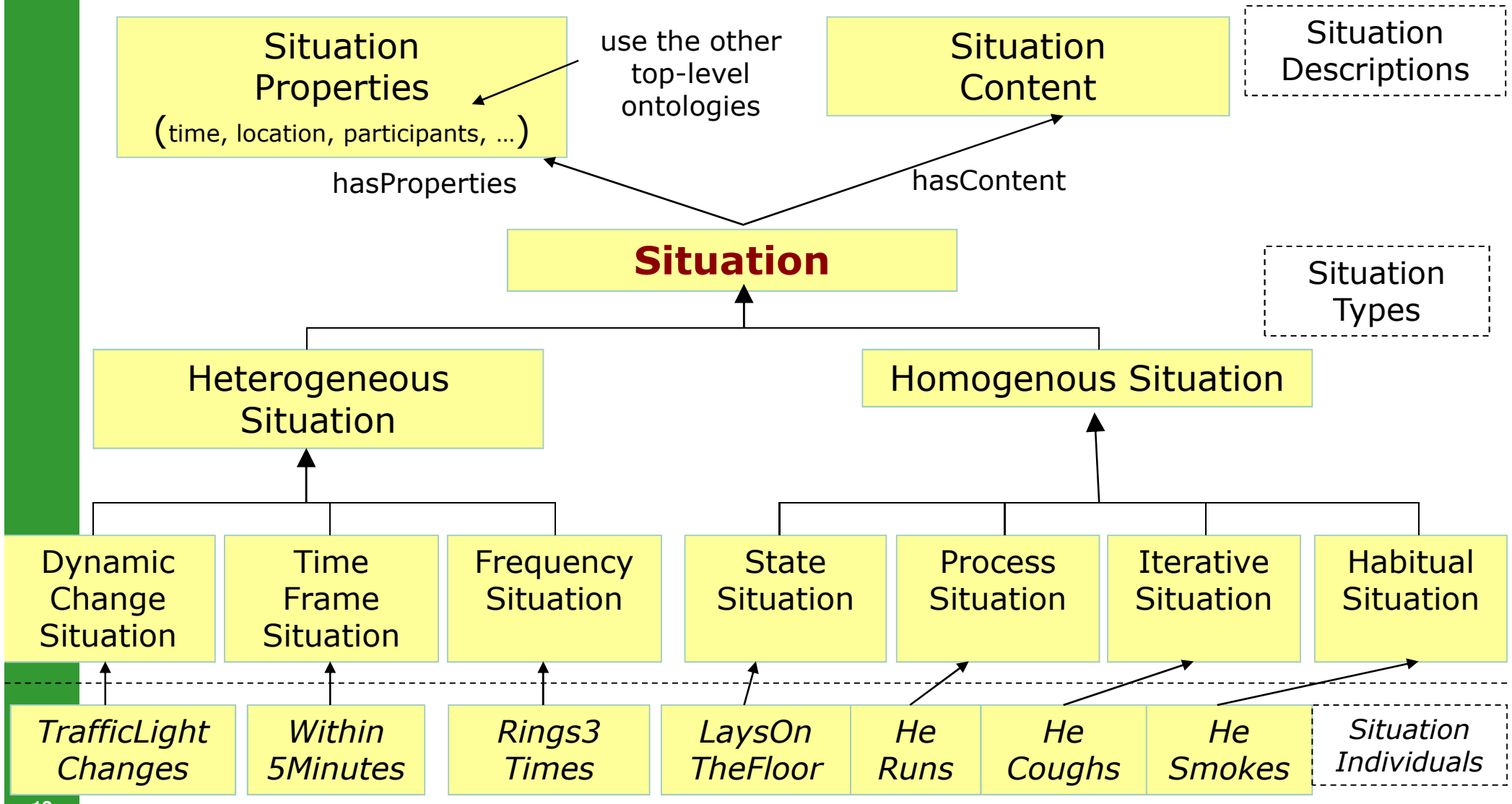
- **CIDOC CRM**: museums and libraries
- **ABC Ontology**: digital libraries
- **Event Ontology**: digital music
- **DOLCE+DnS Ultralite**: event aspects in social reality
- **Event-Model-F**: event-based systems
- **VUevent Model**: An extension of DOLCE and other event conceptualizations
- **IPTC. EventML:** structured event information
- **GEM**: geospatial events
- **Event MultiMedia**: multimedia
- **LODE**: events as Linked Data
- **CultureSampo**: Publication System of Cultural Heritage
- **OpenCyC Ontology**: human consensus reality, upper ontology with lots of terms and assertions
- **Super BPEL**: ontology for the business process execution language
- **Semantic Sensor Net Ontology**: ontology for sensor networks

Modular Ontology Model for SCEP



Source: Kia Teymourian and Adrian Paschke: Towards Semantic Event Processing, DEBS 2009, July 6-9, 2009

Example:
Situation Top-Level Ontology Model



Summary of Ontology Benefits for SCEP

- Ontologies can be applied to CEP application domains as in any other IT system
- Ontologies for time-based events + event operations etc can also improve formalisation of CEP

Rule-based Complex Event Processing

- CEP complex event ***detection*** can use reaction rules
  - e.g. event-condition-action rule:
    incoming event +
    conditions over situations (effects + context) → (re)action
- Detected Complex (aggregated) events and
  their effects (situations) trigger
  reaction rules for decision + behavioral reaction logic
  - e.g. CEP event-condition-action rule:
    complex event + decisioning conditions → response actions
- Rule-based Event Processing Languages (EPLs)
  - e.g. **Reaction RuleML**, IBM Amit Situation Manager Rule,
    TIBCO BusinessEvents, ...

Production Rules and CEP

- Production Rules react to fact changes *(facts = data)*
- However, PR systems typically use an **object model** + events:
  can represent external fact updates — extensible for use in CEP
  - Event types and object classes are defined in the rule declarations
  - New events are added to the fact base / working memory
    - might become an event channel (such as an event queue)
      (facts = events and data)
  - Instance tuples are filtered and joined in the rule conditions
    and — for valid tuples — the action part of the rule is executed
- Can be further extended with mechanisms such as query languages,
  state models and temporal constraints
- Examples: TIBCO BusinessEvents, Red Hat Drools

Event Condition Action Rules and CEP

- ECA Rule

  "*on Event if Condition do Action*"

  – Explicit event part (trigger) + separate data conditions → actions

  – e.g. *on customer order* (event) + *check if credit card is valid* (condition) → *process order* (action)

- Evolved from active databases

  – extend databases with trigger-type reactions,
  e.g. HiPac, Chimera, ADL, COMPOSE, NAOS

  – Composite event algebras, e.g. SAMOS, COMPOSE, Snoop

  – `Sequence | Disjunction | Xor | Conjunction | Concurrent | Not | Any | Aperiodic | Periodic`

Research Standards Area: Reaction RuleML

- **RuleML = family of rule languages in homogeneous interchange format**
  - Technology input to standards bodies like W3C (RIF) and OMG (PRR)
- **Reaction RuleML = general reaction rule format**
  that can be specialized as needed for different rule types (e.g. PR, ECA style rules, …)
  - Platform-independent XML-based **rule interchange format**
    - translation into platform-specific executable rule languages, e.g. Prova
  - Three general **execution semantics**:
    - **Active**: actively 'poll' external event source for events
      e.g. ping a service/system or query an internal or external event database
    - **Messaging**: wait for incoming event from a messaging system
    - **Reasoning**: KR event/action logic reasoning and transitions/transactions
      (e.g. as in Event Calculus, Situation Calculus, Temporal Action Logic formalizations)
  - Appearance
    - **Global**: 'globally' defined reaction rule
    - **Local**: 'locally' defined reaction rule (switched on) for a specific context

General Syntax for Reaction Rules

```
<Rule style="active|messaging|reasoning" eval="strong|weak|defeasible|fuzzy">

      <oid>       <!- object id -->                                  </oid>
      <label>     <!- meta data of the rule -->                      </label>
      <scope><!- scope of the rule e.g. a rule module -->           </scope>
      <qualification> <!- e.g. priorities, validity, fuzzy levels -->
                                                                    </qualification>
      <quantification> <!- e.g. variable bindings-->                </quantification>

      <on>        <!- event part -->                                 </on>

      <if>        <!- condition part -->                             </if>
      <then>      <!- (logical) conclusion part -->                  </then>

      <do>        <!--  action part -->                             </do>

      <after>     <!- postcondition part after action,
                      e.g. to check effects -->                     </after>
</Rule>
```

Reaction RuleML – Example Rule Types

- **Production Rule:**

```
<Rule style="active">
      <if>...</if>
      <do>...</do>
</Rule>
```

- **Trigger Rule:**

```
<Rule style="active">
      <on>...</on>
      <do>...</do>
 </Rule>
```

- **ECA Rule:**

```
<Rule style="active">
     <on>...</on>
     <if>...</if>
     <do>...</do>
</Rule>
```

Reaction RuleML Features

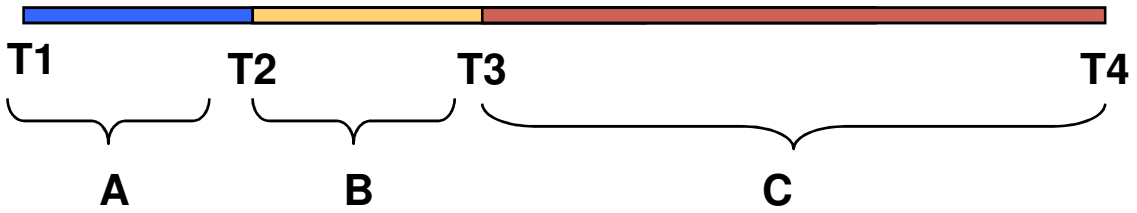
- Action Algebra:
  Succession (Ordered Sequence), Choice (Non-Deterministic Selection), *Flow* (Parallel Concurrent Flow), *Loop* (Iteration)
- Event Algebra:
  Sequence (Ordered), *Disjunction* (Or) , *Xor* (Mutal Exclusion), *Conjunction* (And), *Concurrent* , *Not*, *Any*, *Aperiodic*, *Periodic*
- Event / action messaging
- External data models and ontologies
- Different detection, selection and consumption policies
- Intervals (Time, Event)
- Situations (States, Fluents)
- External event query languages
- ...

Underlying Semantics: Interval-based Event Calculus

- Events initiate and terminate Situations (Fluents) which hold during a time interval
- Interval-based Event Calculus semantics (model-theory + proof theory) based on time intervals modeled as fluents

$$I: Ti \times Fl \rightarrow \{true, false\}$$

- Example: *A;(B;C) (Sequence)*



- Rule-based interpretation of typical complex event detection operators
- Rule-based inference involving times and durations, e.g. Allen's interval logic

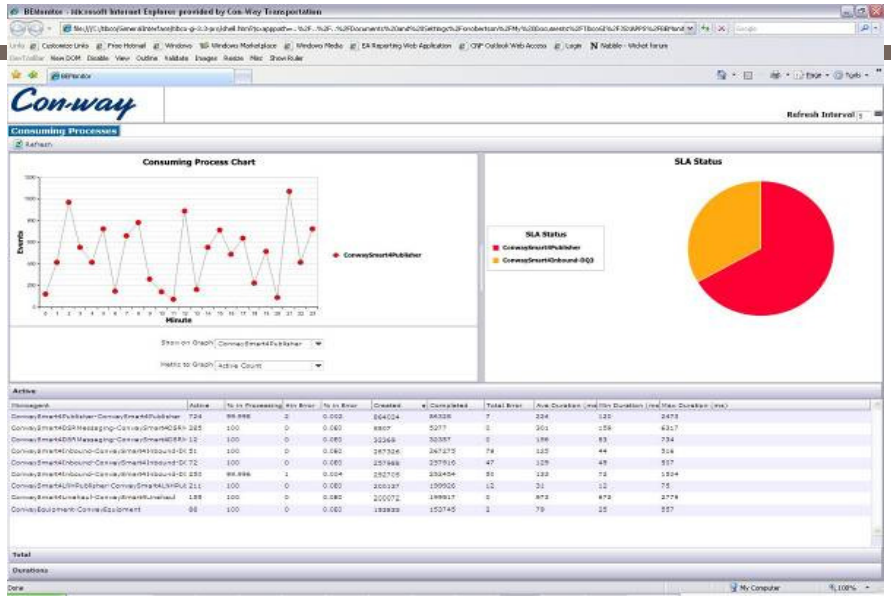
CEP Use Cases

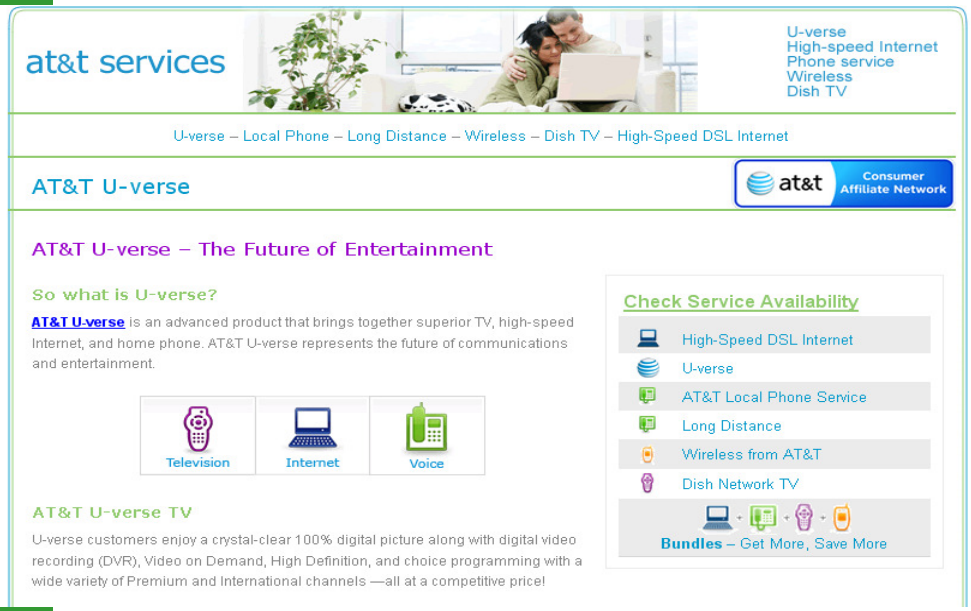
- Current CEP technologies and applications mostly concentrate on performance (vs knowledge, semantics, provable correctness)
- CEP technologies like TIBCO BusinessEvents provide certain level of semantics:
  - UML-level for class, event relationships (inheritance, containment, reference)
  - UML-level for state behaviors
  - BPMN-level for process behaviors
  - UML PRR-level for declarative rule behaviors
  - etc.
- Utilize essentially fixed-schema data representations (e.g. object definitions)
- Machine learning, currently limited to rule / query parameterization (etc.)

Live CEP Use Cases that Involve Semantics (1)

Logistics

- Optimization of shipping movements in transit and in port to obtain greater efficiencies and to reduce fuel costs
- Real-time tracking of cargo, packages border crossing manifests
- Revenue management, baggage handling in airlines





Telco

•Track missing events, SLA violations, re-sequence out-of-order events according to context

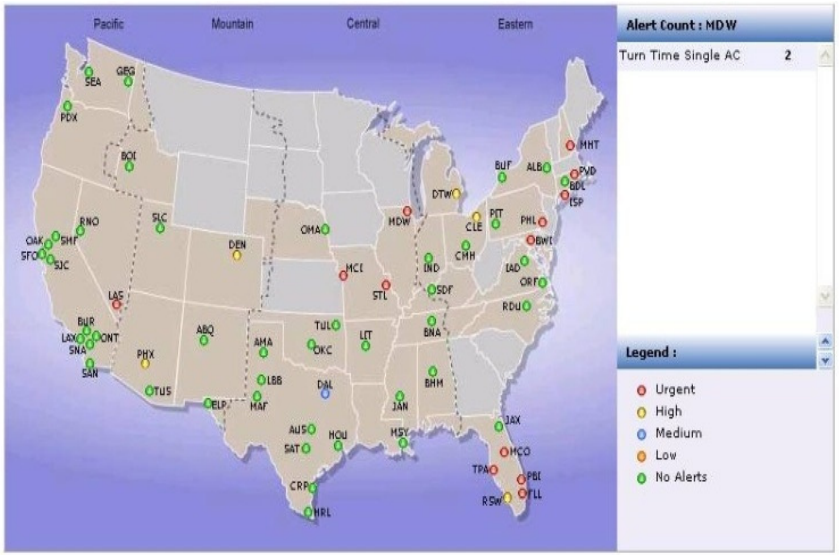
•Identify under-performing business systems to ensure service levels can be met and enhanced

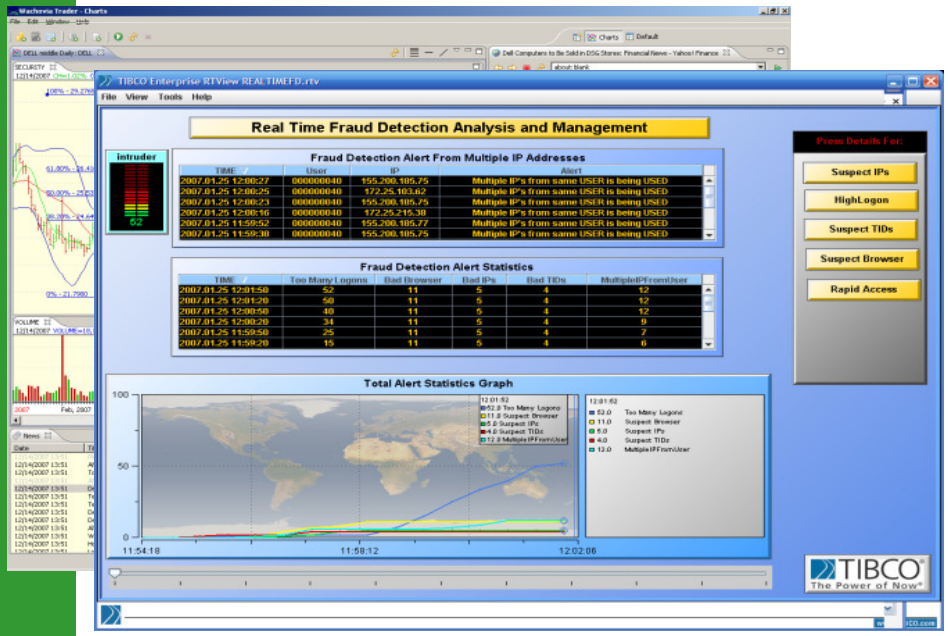
•Location-based services for targeted campaigns

Live CEP Use Cases that Involve Semantics (2)

Energy

- Predictive energy usage / energy event processing
- Smart Grid Initiatives
- Transmission, Outage Intelligence Fault Management



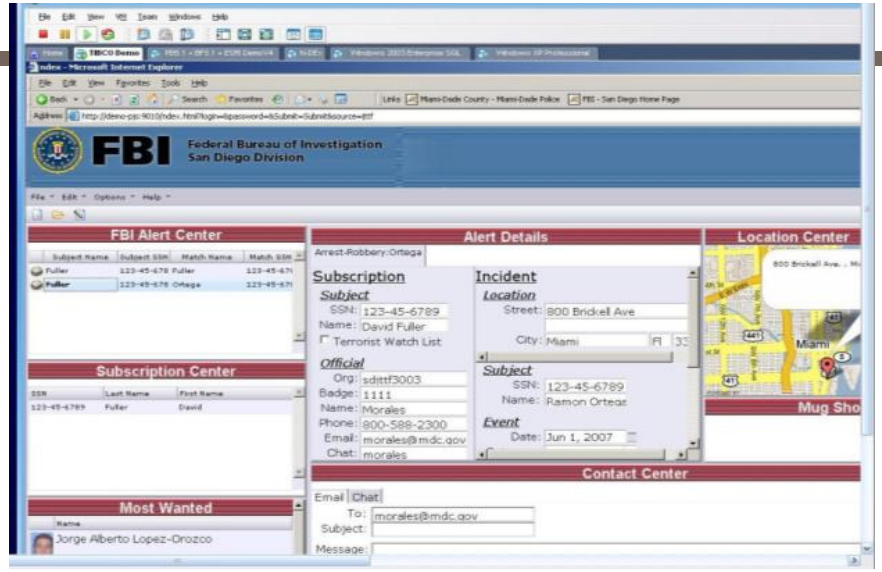


- *Finance*
  Credit derivative trades: real-time workflow and matching of trades
- Visualize market data, order executions
- Fraud / Intrusion Detection
- Track and Trace for Trades/Deals/Settlements and Pre/Post trade exceptions

Live CEP Use Cases that Involve Semantics (3)

Government

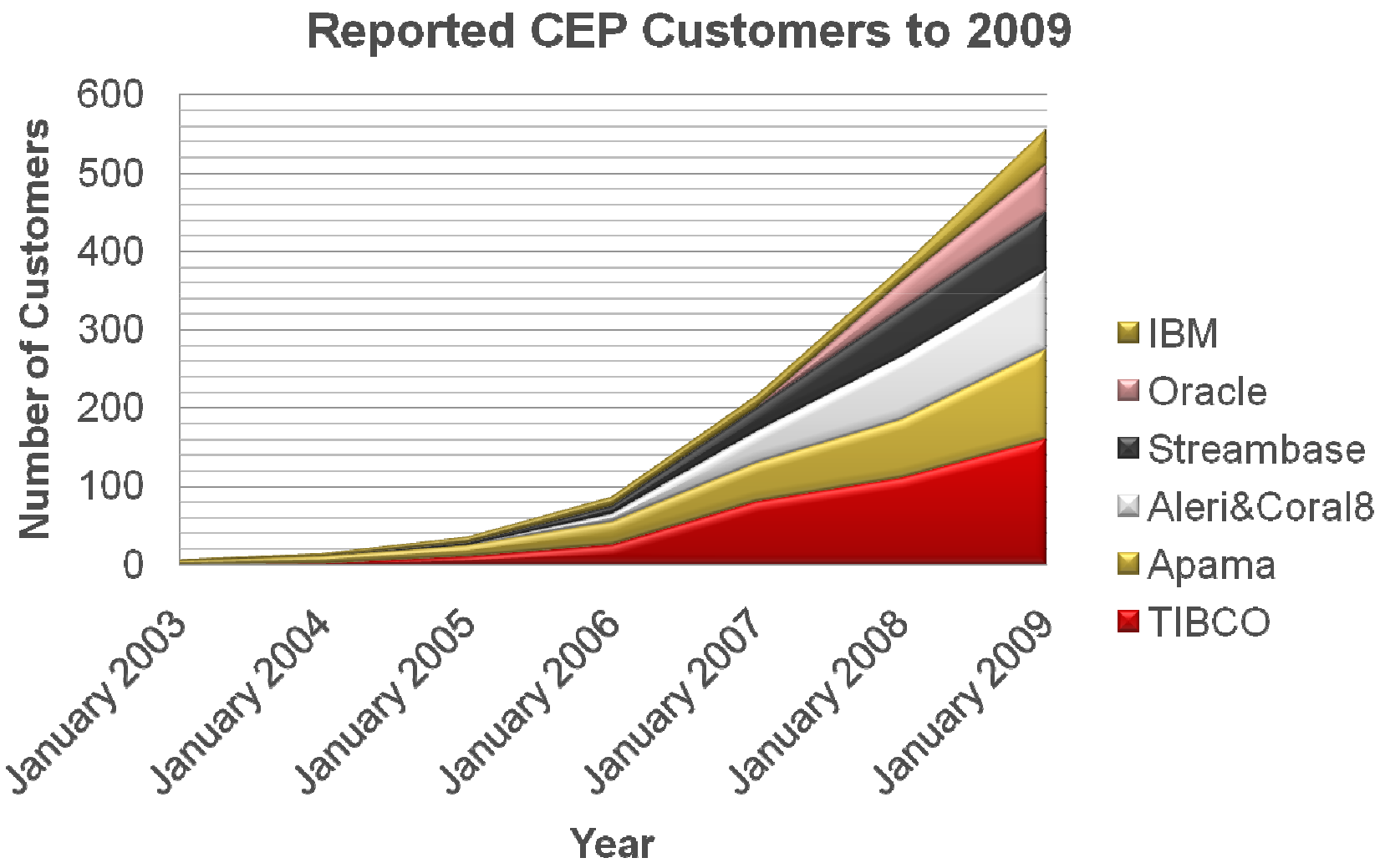
- Track and Analyze ‘patterns’ that were otherwise very difficult to detect
- Real-time tracking of people, places and things
- Capacity planning



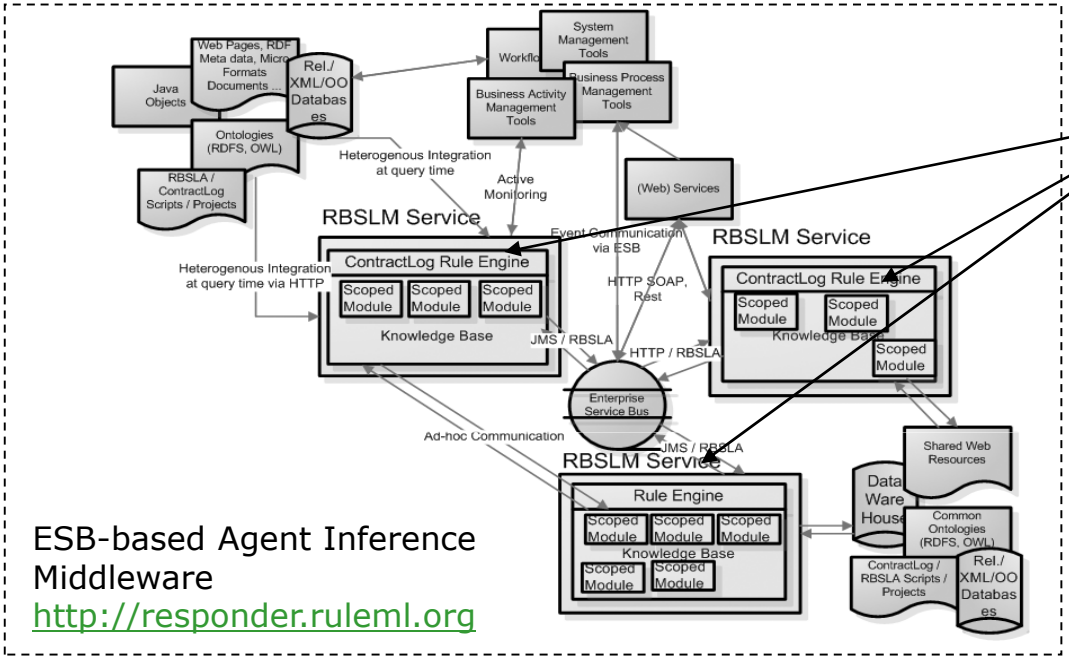
Adaptive Marketing

- Capture opportunity with customer while “window” is open
- Ability to (automatically) learn from successful/failures of campaign in progress to fine tune the offer

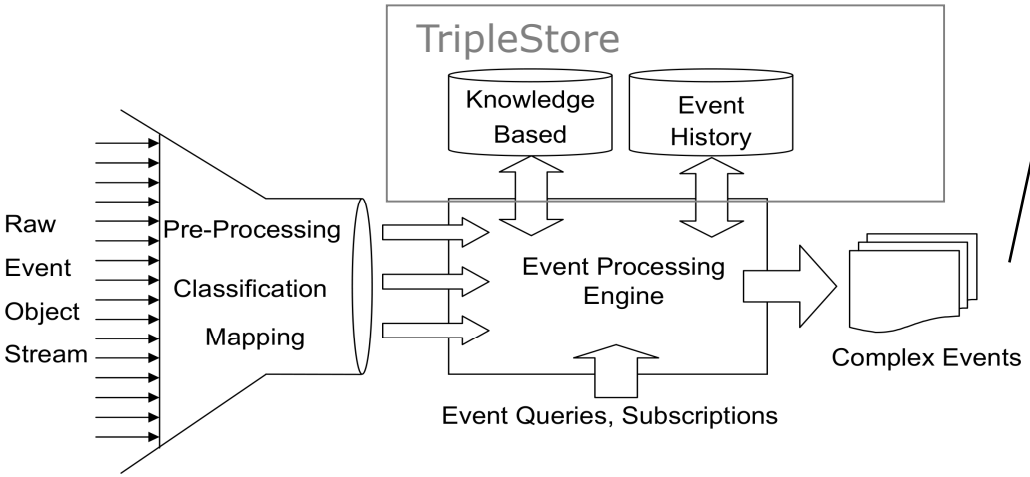
CEP and SCEP as a Software Market

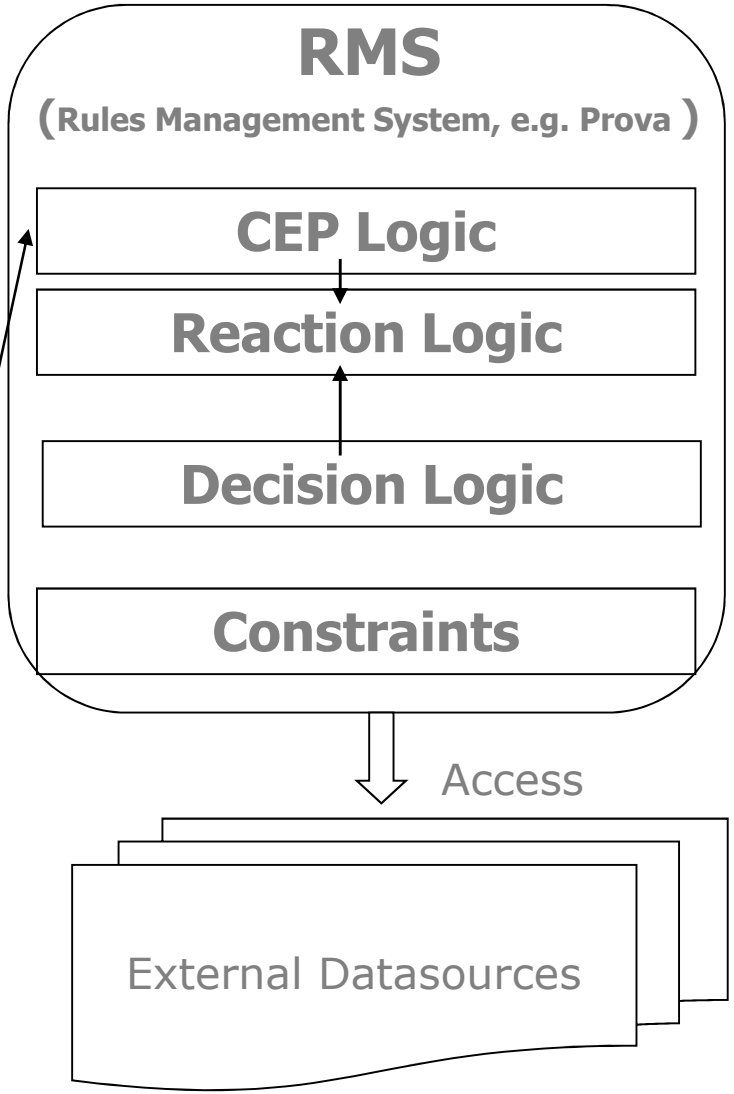


SCEP Implementation via Heterogeneous Integration

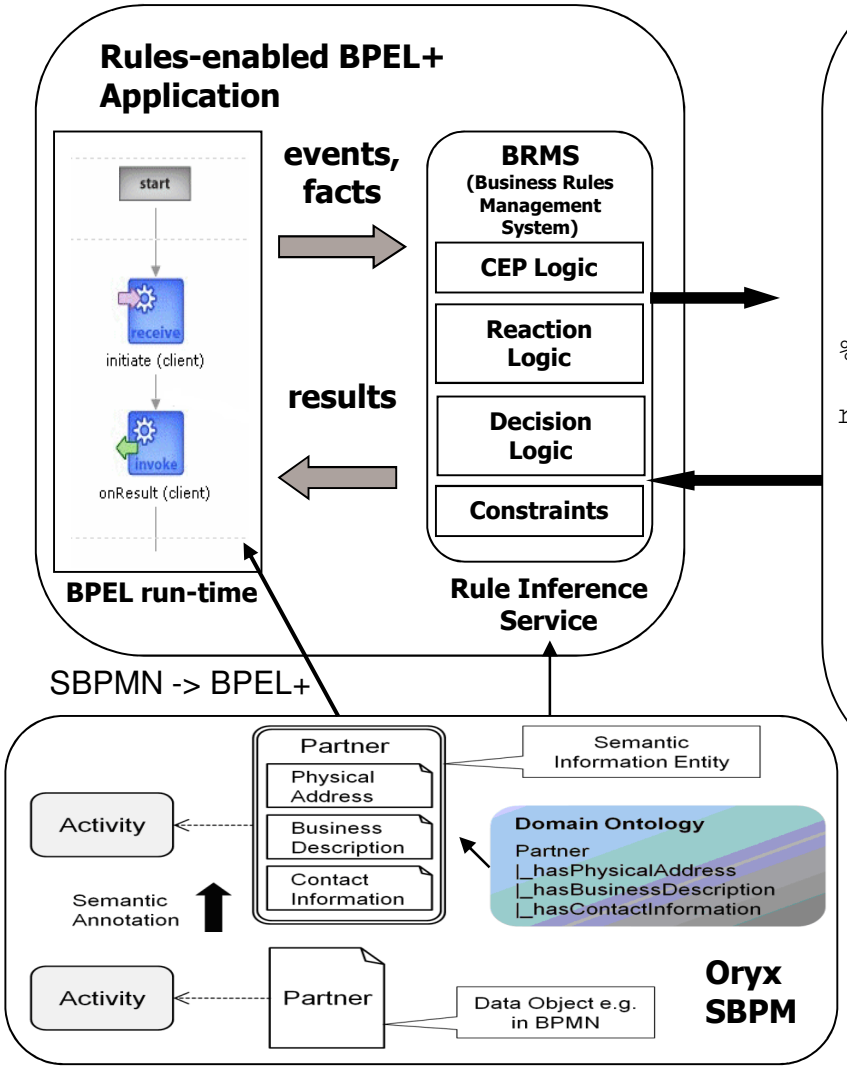


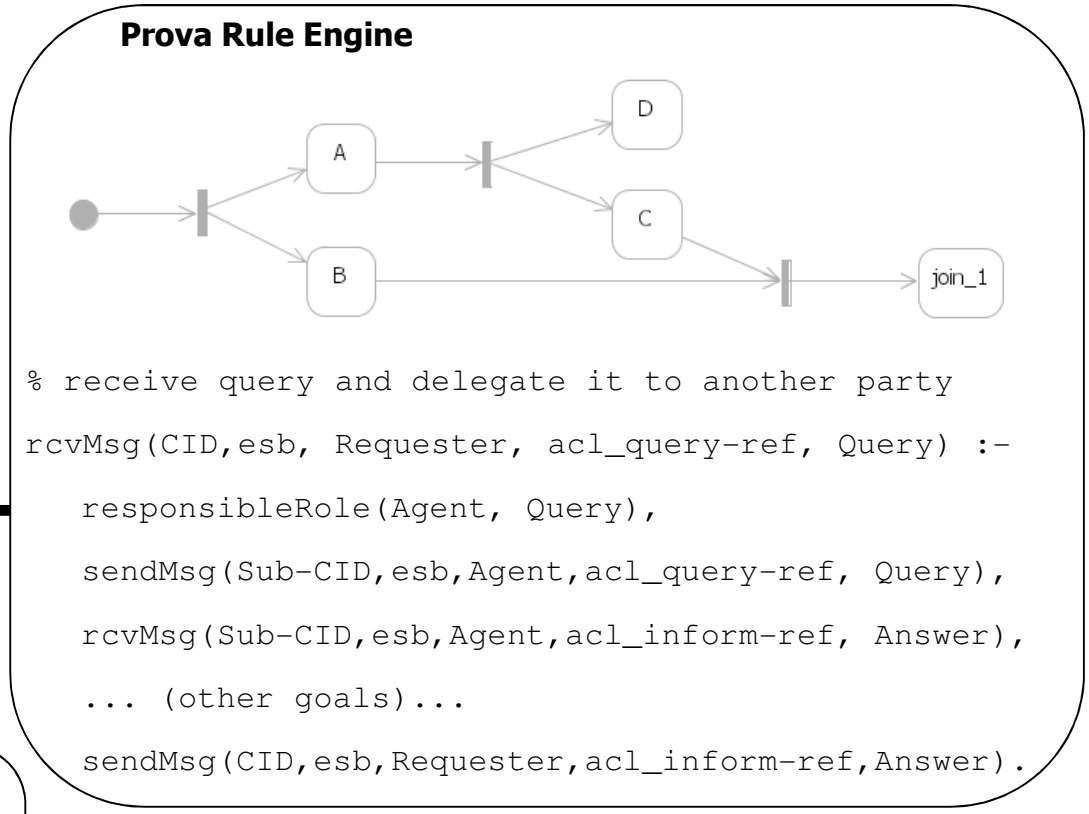
ESB-based Agent Inference Middleware
http://responder.ruleml.org





SCEP Research: Event-Driven Business Process Management (edBPM) BPMN -> Orchestrated BPEL + Choreography Rule Workflow





- Paschke, **Rule Responder BPM / ITSM Project**
- Barnickel, Böttcher, Paschke, **Semantic Mediation of Information Flow in Cross-Organizational Business Process Modeling**, 5th Int. Workshop on Semantic Business Process Management at ESWC 2010
- Adrian Paschke and Kia Teymourian, **Rule Based Business Process Execution with BPEL+ ,** i-Semantics 2009, Graz
- Paschke, A., Kozlenkov, A.: **A Rule-based Middleware for Business Process Execution**, at **MKWI'08**, München, Germany, 2008.

Summary for Semantic Complex Event Processing

- Event processing engines **understand** what is happening in terms of (top-level and domain) ontologies
- Reaction rule engines **know** what (re)actions and processes they can invoke and what events they can signal
- Makes it possible to detect and **process semantically related (complex) events**
- Needs **formal semantics** for concepts relevant in event processing, e.g. time, spatio, state, action behavior …
  - also support interchange across domain boundaries with different vocabularies
- Many use cases in different industrial domains demonstrate the **usefulness** of combining **Semantics** (Rules & Ontologies) with **CEP**

Thank you …