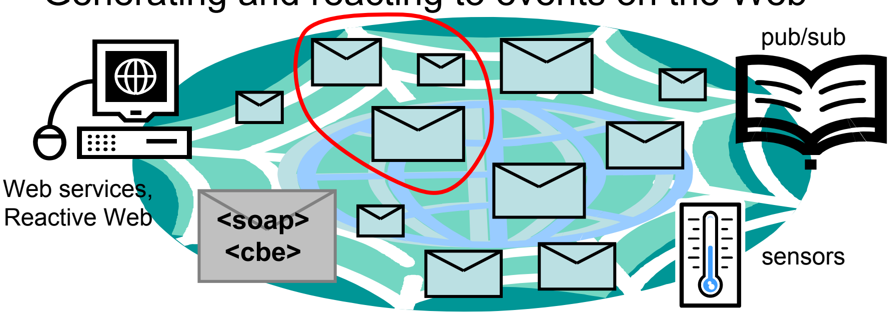
Rule-Based Composite Event Queries: The Language XChange^{EQ} and its Semantics

François Bry, <u>Michael Eckert</u>
LMU München

Motivation: Composite Events

Generating and reacting to events on the Web

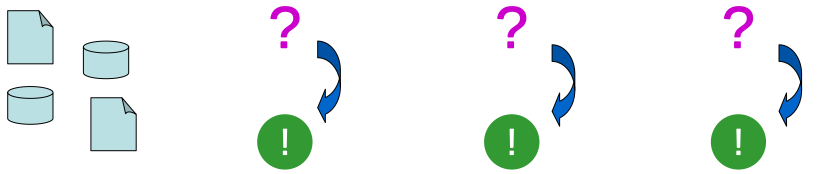


Composite Events

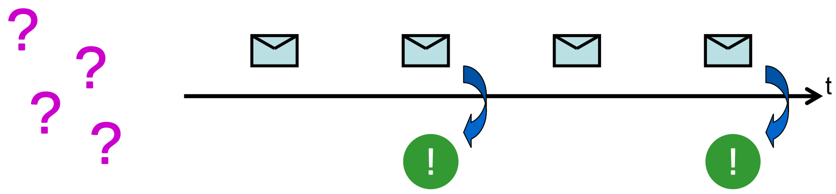
- Must be inferred from “atomic events” (messages)
- Multiple atomic events, relationship between them
- Need query language!

Queries against Event Streams

Database Queries, Web Queries:

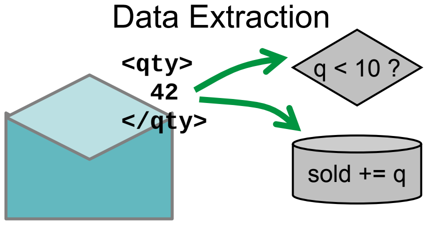


(Composite) Event Queries:

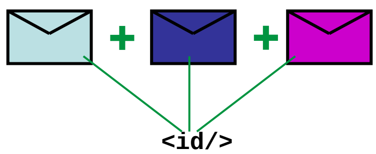


Language Requirements

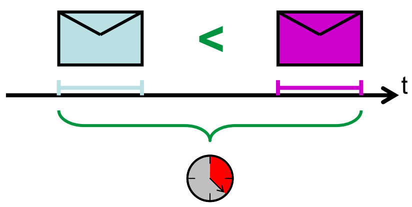
Data Extraction



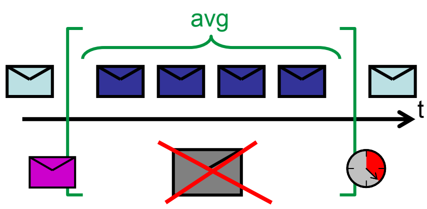
Event Composition



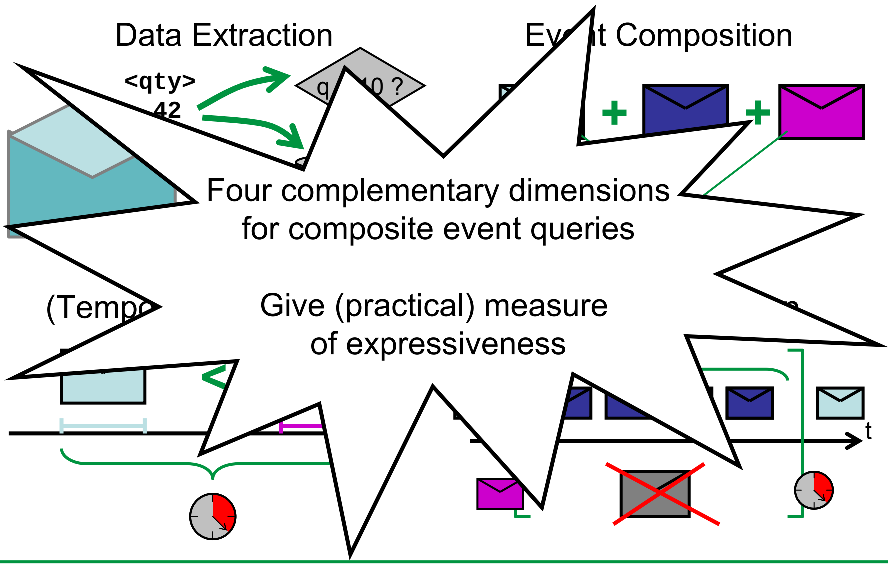
(Temporal) Relationships



Event Accumulation



Language Requirements



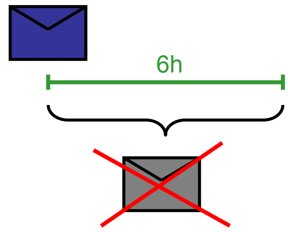
XChange^{EQ}: Rule-Based CEQs

- High-level, declarative query language for composite events, fully covers four dimensions
- Pattern-based queries on XML event messages: embeds Web query language Xcerpt
- Integrates into reactive rule language XChange
  - Perform automatic reactions, timing important
- Deductive (event) rules:
  - Define new, “virtual” events from received events
  - mediation, abstraction, reasoning (cf. database views)
  - Side-effect free; don’t implement by reactive rules: optimization, (human) understanding

XChange^{EQ}: Example Rule

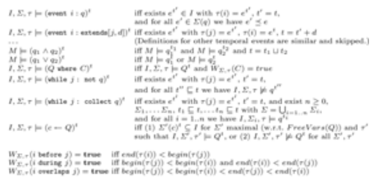
```
DETECT
  overdue {
    id { var ID }
    cust { var C } }
ON
  and {
    event o: order {{
      id { var ID },
      quantity { var Q },
      cust { var C } }}
    event w: extend[o, 6h],
    while w: not shipped {{
      id { var ID } }}
  } where { var Q < 10 }
END
```

```
DETECT
  order {
    id { var ID},
    quantity { var C },
    cust { var C } }
FROM
  ...
END
```



Semantics (1)

- Declarative Semantics for XChangeEQ: *model + fixpoint theories for stratified programs* (A standard approach for rule languages)
- (Tarski-style) model theory: Define M ⊨ F^t recursively



- Accommodates event identifiers (“event o:”)
- Events have occurrence times
- Temporal relations: fixed interpretation

Semantics (2)

- Restriction to stratified programs
  - w.r.t. negation, grouping, *relative temporal events*
- Fixpoint: model $M_{P,E}$
  - $T_P(I)$: all events derivable by rules in P
  - starting with incoming event stream E
  - compute fixpoints stratum by stratum
- Theorem:
  - P stratified program, E (incoming) event stream.
  - Then: $M_{P,E}$ is a minimal model of P under E and
  - Independent of the stratification of P

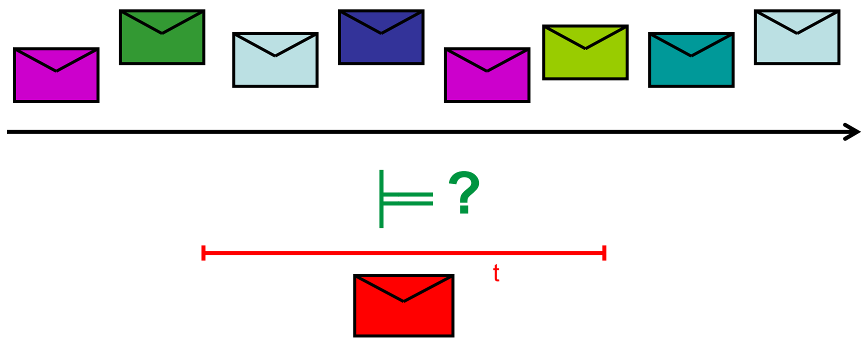
$$P = P_1 \uplus \cdots \uplus P_n$$

$$T_P(I) = I \cup \{e^t \mid \text{there exist a rule } c \leftarrow Q \in P, \text{ a maximal substitution set } \Sigma, \text{ and a substitution } \tau \text{ s.t. } I, \Sigma, \tau \models Q^t \text{ and } e \in \Sigma(c)\}$$

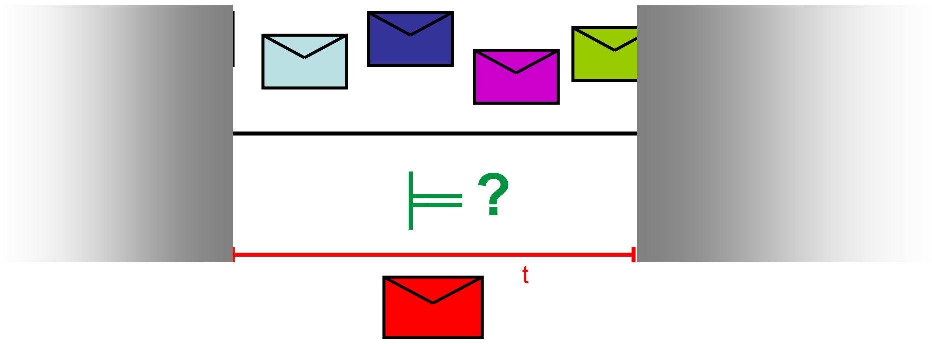
T_P^ω: least fixpoint of T_P

$$M_0 = E = T_\emptyset^\omega(E), \quad M_1 = T_{P_1}^\omega(M_0), \quad \ldots, \quad M_n = T_{P_n}^\omega(M_{n-1}) =: M_{P,E}.$$

Unbounded Event Streams



Unbounded Event Streams



- XChange^{EQ}-Semantics are well-defined for *unbounded* ("infinite") incoming event streams E

$$M_{P,E} \mid t = M_{P,E|t} \mid t$$

Summary and Outlook

- XChange^{EQ}:
  - High-level event query language
  - Full coverage of all four dimensions, XML support
  - Support for (deductive) event rules
- Declarative Semantics
  - Model and fixpoint theory for stratified programs
  - Well-defined on unbounded event streams
- Outlook
  - Incremental, data-driven evaluation
  - Optimizations based on temporal conditions