ACTrESS – Automatic Context Transformations in Event-based Software Systems

DEBS 2012

Tobias Freudenreich
Stefan Appel
Sebastian Frischbier
Alejandro Buchmann
How we picture event-based systems

Event Bus

Event Bus
How we picture event-based systems
What is the problem with that?

Event producers and consumers are all over the place

What does it mean if an event notification reads

weight = 25

Interpretation adds meaning to data!
Why this is more than data integration

There are commercial products for data integration in databases

But: EBS have some key additional challenges

anonymity  
dynamics  
latencies important
The idea of ACTrESS
The idea of ACTrESS

Broker

Connection Handler

Message Handler

transforms data
ACTrESS uses interpretation contexts

Producers and consumers pass their interpretation context to the broker network
Defining contexts

Types

- DeliverySpecs
  - weight [kg]
  - Address [addr]
  - firstName
  - lastName
  - specifics

Rules

Functions

- meters-to-yards
- meters-to-feet
- kg-to-lb

---

Types

- USDeliverySpecs
  - weight [lb]
  - USAddress [USad]
  - firstName
  - lastName
  - number
  - ...

Rules

- weight ➔ kg-to-lb
- Address ➔ toUSAddress

Functions

- toUSAddress = {
  ...  
}

7/19/2012  |  Dept. of CS  |  Databases and Distributed Systems  |  Tobias Freudenreich  |  9
The different flavors of transformation

- **unit conversion**
- **check signature**
- **currency conversion**
- **adding position**

**Internal knowledge**
- pure
- stateful

**External knowledge**
- lookup
- “enrichment”
The architecture of ACTrESS
ACTrESS features

- extensible
- flexible
- reusable contexts
- no globally accepted schema required
ACTrESS implementation

- did not have to recompile ActiveMQ
- intercept each incoming and outgoing message
- separate channel for administrative messages
Benchmark Setup

Compared four scenarios:
- channel-based
- content-based
- ACTrESS
- self-describing
Testing the performance

latency [μs]

consumers/producer

self-describing

ACTrESS

content-based

channel-based

1000

2000

2850
Testing the performance

- Channel-based
- Content-based
- ACTrESS
- Self-describing

![Graph showing latency (ms) vs. events/second for different types of systems: channel-based, content-based, ACTrESS, and self-describing. The graph indicates that self-describing systems have significantly higher latency compared to channel-based and content-based systems.](image-url)
Conclusion

- Anonymity in EBS causes interpretation challenges
- ACTrESS enables producers and consumers to specify their context
  - allows for correct matching
  - notifications are transformed transparently
  - modifiable at runtime
- Implementation builds upon an industry-strength product

Anonymity → Data Interpretation Challenge → ACTrESS