Deriving a Unified Fault Taxonomy for Event-Based Systems

Waldemar Hummer, Christian Inzinger, Philipp Leitner, Benjamin Satzger, Schahram Dustdar

Distributed Systems Group, Vienna University of Technology

6th ACM International Conference on Distributed Event-Based Systems (DEBS)

Berlin (Germany)
July 18, 2012
Motivation

- **Dependability and Fault Tolerance are Key Requirements**
  - For business- and safety-critical applications
  - Dictated by business regulations, contractual agreements, laws, ...
  - Many research directions have common understanding of faults, e.g.:
    - Program and operating system faults [3]
    - Faults in object-oriented software [10]
    - Faults in service-based applications [11, 14]
    - Faults in memory devices [70]

- **Emerging Field of Event-Based Systems (EBS)**
  - Different sub-areas and specializations of EBS, e.g.:
    - Complex Event Processing (CEP),
    - Event Stream Processing (ESP),
    - Wireless Sensor Networks (WSNs),
    - Event-Driven Programming, etc.
  - Many commonalities, yet: **No common understanding of potential faults and fault sources in EBS**
Contribution Outline

- Identification of 5 Main Sub-Areas of EBS
  - Extensive Literature Review (75+ papers) of the Past Two Decades

- Definition of a Common Model for EBS
  - Compiled From Previous Publications, e.g.:
    - *A Conceptual Model for Event Processing Systems* (Moxey et al., 2010)
    - *Towards a Common Event Model for an Integrated Sensor Information System* (Fowler and Qasemizadeh, 2009)
    - *Toward a Common Event Model for Multimedia Applications* (Westermann and Jain, 2007)

- Deriving a Fault Taxonomy for EBS
  - Based on Highly Influential Previous Work (Avizienis et al., 2004)
  - Taxonomy Dimensions: Fault Classes and Fault Sources
  - Discussion of 30 Fault Instances

- Discussion of Application Possibilities
  - Fault Injection
  - Fault Analysis
Sub-Areas of EBS

- Event-Driven Interaction Paradigms
  - Event-Driven Programming (EDIP)
  - Event Stream Processing (ESP)
- Event-Driven Business Process Management (EDBPM)
  - Pub/Sub
    - Producers
    - Consumers
    - Subscriptions
    - Events
    - Nodes
    - Operations
- Wireless Sensor Networks (WSN)
  - Complex Event Processing (CEP)
    - Power Constraints
    - Dynamic Topologies
    - Event Routing
  - Monitoring
  - Process Mining
  - Process Instances
- Complex Event Processing (CEP)
  - Analytics & Prediction
  - Activity Detection
  - Load Shedding
  - Approximate Queries
  - High Throughput
Sub-Areas of EBS

- Various Relationships Between Sub-Areas
  - Share Many Commonalities
  - Different Terminology for Similar Concepts

<table>
<thead>
<tr>
<th>Concept [50]</th>
<th>EDIP</th>
<th>ESP</th>
<th>CEP</th>
<th>WSN</th>
<th>EDBPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>notification</td>
<td>tuple</td>
<td>event</td>
<td>datum</td>
<td>invocation</td>
</tr>
<tr>
<td>producer</td>
<td>publisher</td>
<td>source</td>
<td>producer</td>
<td>sensor</td>
<td>service/activity</td>
</tr>
<tr>
<td>consumer</td>
<td>subscriber</td>
<td>sink</td>
<td>consumer</td>
<td>sink</td>
<td>service bus</td>
</tr>
<tr>
<td>event processing</td>
<td>service</td>
<td>operator</td>
<td>agent</td>
<td>node</td>
<td></td>
</tr>
<tr>
<td>channel</td>
<td>channel</td>
<td>stream</td>
<td>event bus</td>
<td>link</td>
<td>service bus</td>
</tr>
<tr>
<td>derived event</td>
<td>merged message</td>
<td>event pattern</td>
<td>complex event</td>
<td>fused information</td>
<td>composite service</td>
</tr>
</tbody>
</table>

- Idea: Explicitly Combine and Integrate Different Views
  - Identification of potential issues and faults
    
    Diverse Goals and Specializations
    
    Potential Faults on Various Conceptual Levels

- Step towards a better understanding of EBS
Common Model for EBS

- Based on the terminology of Moxley et al [50], e.g.,
  - Event Processing Agent (EPA)
  - Event Processing Network (EPN)
  - ...
Common Model (2)

- **Basic Model Artifacts**
  - Event, Subscription, Producer, Consumer, …
  - Three Core Interfaces
    - Subscription, Notification, Publication

- **Artifacts to Model Technical Details**
  - Buffer (for event input and output)
  - State
  - Input-Output Operator Function
  - Deployment of EPAs

- **Artifacts for Advanced Concepts**
  - Complex Event
  - Correlation
  - Dependency
EPA = Event Processing Agent
Behavior of EPAs defined by two types of functions

- **Input-Output Operator Function**
  - \( \text{op}: (BE \times S) \rightarrow (BE \times S) \)

- **Event Routing Functions**
  - \( \text{in}: (E \times P \times S) \rightarrow \mathcal{P}(B) \)
  - \( \text{out}: (E \times \mathcal{P}(B) \times S) \rightarrow \mathcal{P}(C) \)

**Variables**
- \( E \) ... Domain of Possible Events
- \( P \) ... Set of Event Producers
- \( C \) ... Set of Event Consumers
- \( B \) ... Set of Event Buffers
- \( S \) ... State of the EPA
- \( \mathcal{P}(X) \) ... Powerset of \( X \)

**Acronyms**
- EPAs: Event-Driven Process Automata
Fault Taxonomy

- **14 Elementary Fault Classes**
  - What are fundamental characteristics of the faults?
  - Based on Highly Influential Work by Avizienis et al. [5]
    - Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 2004

- **19 Fault Sources**, Grouped into 6 Categories
  - Which elements of the system are responsible for causing the fault?
  - Closely based on the artifacts in the common model for EBS

- **30 Distinct Fault Instances**
  - What are concrete manifestations of the fault taxonomy in real-life event-based systems?
  - Derived along the Fault Classes Taxonomy
  - Attributed to the respective Fault Sources
Fault Classes

- Phase of Creation or Occurrence
  - Development Faults
  - Operational Faults
- Level in Solution Stack
  - Platform Faults
  - Business Logic Faults
- System Boundaries
  - Internal Faults
  - External Faults
- Persistence
  - Permanent Faults
  - Transient Faults
- Dimension
  - Hardware Faults
  - Software Faults
- Phenomenological Cause
  - Natural Faults
  - Human-Made Faults
- Capability
  - Accidental Faults
  - Incompetence Faults

- Differences to Avizienis et al. [5]:
  - Additional: Level in Solution Stack
  - Left Out: Malicious/Non-Malicious and Deliberate/Non-Deliberate (security-related and not as relevant for dependability)
Fault Sources

- Six Categories of Fault Sources
  - compiled from earlier work on fault localization [3, 63] and root cause analysis [38]
Identified Fault Instances

Taxonomy Tree based on the 14 Fault Classes

- Perm ... Permanent
- Trans ... Transient
- HW ... Hardware
- SW ... Software
- Na ... Natural
- Hu ... Human-Made
- Ac ... Accidental
- In ... Incompetence
Identified Fault Instances (2)

- Factors of Influence Responsible for Different Faults
- Grid of Fault Sources (rows) and Fault Instances (columns)
  - Red dots mark the intersection of faults and factors of influence (i.e., potential fault sources)
Fault Examples

- **Fault: Buffer Overflow**
  - **Problem Manifestation**
    - Attempt to store more events than available (free) buffer slots.
  - **Main Affected Model Artifacts**
    - EPAs, Event Buffers, Input-Output Operator Function
  - **Fault Classes**
    - Phase of Creation or Occurrence: Operational
    - Level in Solution Stack: Platform
    - System Boundaries: External
    - Persistence: Transient
    - Dimension: Software
    - Phenomenological Cause: Human-Made
    - Capability: Accidental
  - **Potential Fault Sources**
    - Environment, External Inputs, Code Functions, System State, Software Assets
Fault: **Outdated Subscription**

- **Problem Manifestation**
  - Subscription with past expiry date has not been garbage-collected.
  - (comparable to a *dangling pointer* in programming languages)

- **Main Affected Model Artifacts**
  - EPAs, State, Subscriptions

- **Fault Classes**
  - Phase of Creation or Occurrence: Development
  - Level in Solution Stack: Platform
  - System Boundaries: Internal
  - Persistence: Permanent
  - Dimension: Software
  - Phenomenological Cause: Human-Made
  - Capability: Incompetence

- **Potential Fault Sources**
  - External Inputs, System State
Fault: **Transmission Bit Flip**

- **Problem Manifestation**
  - Event messages are unintentionally modified when transmitted over a (unreliable) network link.

- **Main Affected Model Artifacts**
  - EPAs, Event, Event Properties, Input-Output Operator Function

- **Fault Classes**
  - Phase of Creation or Occurrence: **Operational**
  - Level in Solution Stack: **Platform**
  - System Boundaries: **External**
  - Persistence: **Transient**
  - Dimension: **Hardware**
  - Phenomenological Cause: **Natural**
  - Capability: **Accidental**

- **Potential Fault Sources**
  - Environment
Application Possibilities

- **Huge Potential for Practical Applications**
  - Dependable Operation of EBS Poses Great Challenges
  - Exciting Research Questions, Focus of Our Ongoing Work

- **Two Main Orthogonal Goals**
  - Fault Diagnosis [30] → Fault Detection, Fault Isolation, ...
  - Fault Injection [31,32] → Fault Tolerance

- **Approach Outline**
  - Encode the UML Model in a Software Tool
  - Model and Monitor Runtime Aspects of the Target Platform
    - „Models@run.time“ approach (e.g., Blair et al.)
  - 2-Way Synchronization
    - Keep the model in sync with the real system
    - Allow changes in the model to be reflected in the real system
  - Detect/Predict Faults (Observation) & Inject Faults (Manipulation)
Application Outlook

- Rough Architectural Sketch (Future Work)

```
Fault Simulation and Diagnosis Tool

Model-to-System Adapter

EBS Platform 1

Model-to-System Adapter

EBS Platform 2

Runtime Models

2-Way Synchronization
```
Conclusion

- The Need for a Common Understanding of Faults
  - EBS is a Heterogeneous Research Field
    - Different Specializations, Yet Many Commonalities
  - Most research areas generated comprehensive fault taxonomies
  - So far, no common model of faults in EBS

- A Step Towards a Fault Taxonomy for EBS
  - Identification of 5 Main Sub-Areas of EBS
  - Common Model for EBS
    - Integrates the Requirements of Various Previous Publications
  - Taxonomy Derived Along 14 Fault Classes, 6 Types of Fault Sources
  - Discussion of 30 Concrete Fault Instances

- Exciting Future Research Directions
  - Implementation of the Model and Fault Taxonomy
  - Fault Diagnosis and Fault Injection
  - Collaborative Modeling of Fault Taxonomies
    - Integrate more viewpoints and sub-areas into the model
Discussion
References (1)


References (3)


References (5)