

# Looking for Patterns in Content: From Design to End-Users Consumption.

Panel discussion, CONTENT 2011, Rome, Italy.

Hans-Werner Sehring, T-Systems Multimedia Solutions GmbH.

Erleben, was verbindet.



# Patterns for and in Content?

- In content management patterns can be applied **at all layers**.
  - Software.
  - Models.
    - Content model.
    - Navigation model.
    - Context model.
    - ...
  - Content.
- **Here:** patterns for content.
  - For content analysis / schema derivation.
  - For content structures.
  - For content representation.



# On the Notion of Pattern.

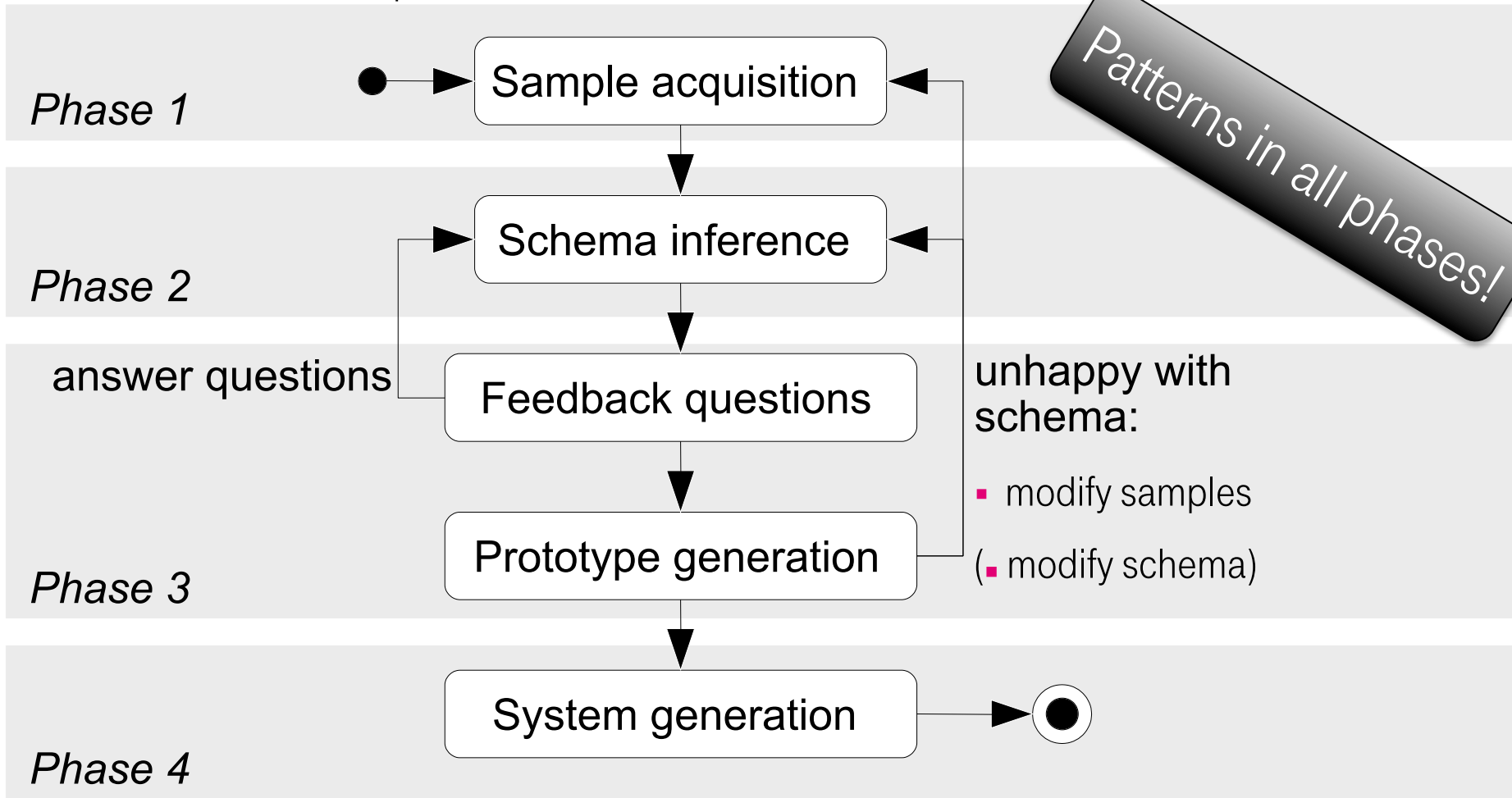
- **What is a pattern in the first place?**
- Two ways to look at them:
  - Predefined abstract solutions to recurring problems; in the sense of Alexander.
  - Recurring structures observed in objects; in the sense of pattern matching approaches.
- **What does it mean for content management?**
- According to the two views from above:
  - Definition of patterns for typical cases of content utilization.
  - Detection of patterns while analyzing content for, e.g., model building and content syndication.



# Patterns for Content Analysis.

## Example: The Asset Schema Inference Process (ASIP).

- The ASIP has four phases:



# Patterns for Content Analysis (cont'd).

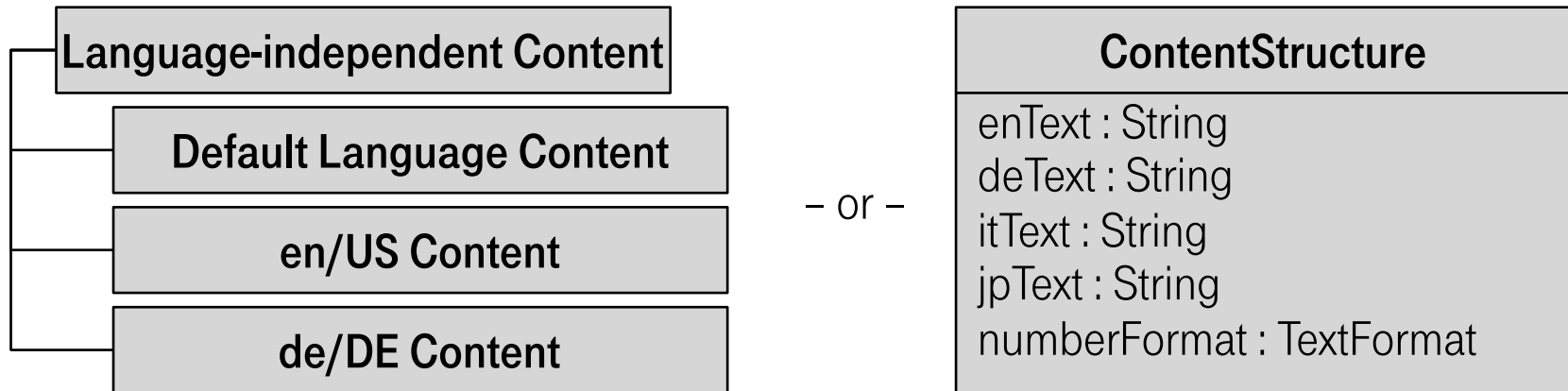
## Two Schema Inference Experiments.

- Experiments with alternatives for **phases 2 and 3**:
  - **(Traditional) schema inference plus user feedback.**  
Straight-forward approach starting from singletons.
  - **Clustering, supervised by domain experts.**  
Statistical approach, semi-supervised learning.
- **Phase 3** (generation of **questions** to gather **feedback**) is determined by the alternative chosen.
- Result of phases 1-3 is a CCM model:
  - **Prototype generation** and **system generation (phase 4)** are carried out by the CCM model compiler.
  - The domain expert can modify the inferred schema (openness and dynamics).



# Patterns for Contextualized Content and Content Use.

- Patterns for typical **content utilization**.
- Currently there are no best practices for recurring problems.
- **Example:** I18n.



- There seems to be no pattern catalogue for these kinds of challenges.

**Pattern definitions could be a tool that helps designing content structures support typical problems in a more adequate way.**



# Patterns for Content Representation.

- **Computer science:**  
Processing of symbols that represent entities (of the real world).
- **Application areas:**

Abstraction level	Symbols	Processing
<b>Computing</b>	Numerals with natural meaning	Evaluation of expressions
<b>Data management</b>	(Domain) Data in standardized form	Standardization, Maintenance, Communication
<b>Content / knowledge management</b>	Multimedia content and subject structures relevant for a specific domain	Context-dependent descriptions and communication



**Typical approach:** Reduction on lower level.



Thank you. Let's discuss.





# Building Software Applications from Software Architectural Patterns

Hassan Gomaa  
*Dept. of Computer Science  
George Mason University  
Fairfax, Virginia, USA  
hgomaa@gmu.edu*

**PANEL**  
**Third International Conferences on  
Pervasive Patterns and Applications  
(PATTERNS 2011)**  
**Rome, Italy**  
September 29, 2011

Copyright © 2011 Hassan Gomaa



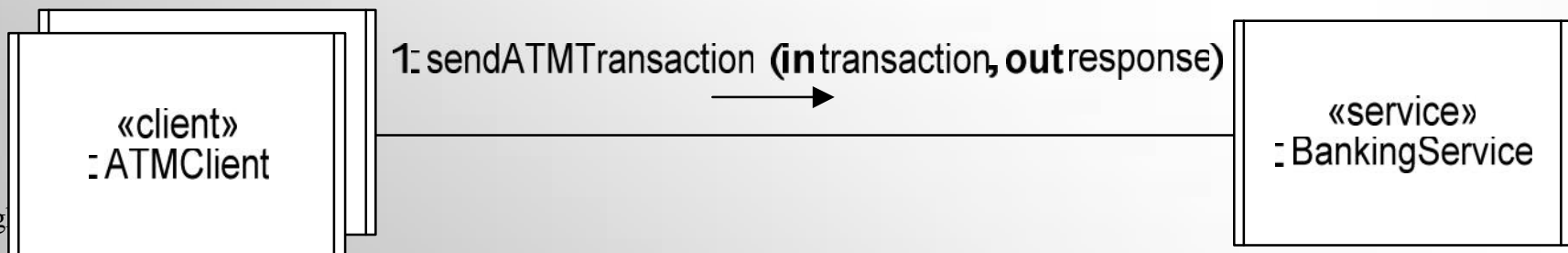
# Software Architectural Patterns

- Software Architectural Patterns [Buschmann, Shaw]
  - Recurring architectures used in various software applications
- Goal: Design Software Architecture from
  - Software Architectural Patterns
- Architectural Structure Patterns
  - Address structure of major subsystems
- Architectural Communication Patterns
  - Reusable interaction sequences between components
- H. Gomaa, *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures*, Cambridge University Press, 2011

# Architectural Structure Patterns

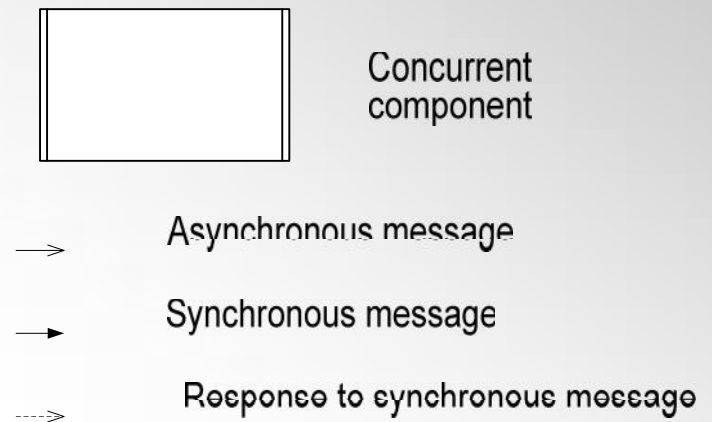
- Layered patterns
  - Layers of Abstraction
- Client/Service patterns
  - Multiple Client / Single Service
  - Multiple Client / Multiple Service
  - Multi-tier Client / Service
- Control Patterns
  - Centralized Control
  - Distributed Control
  - Hierarchical Control

## Multiple Client / Single Service pattern

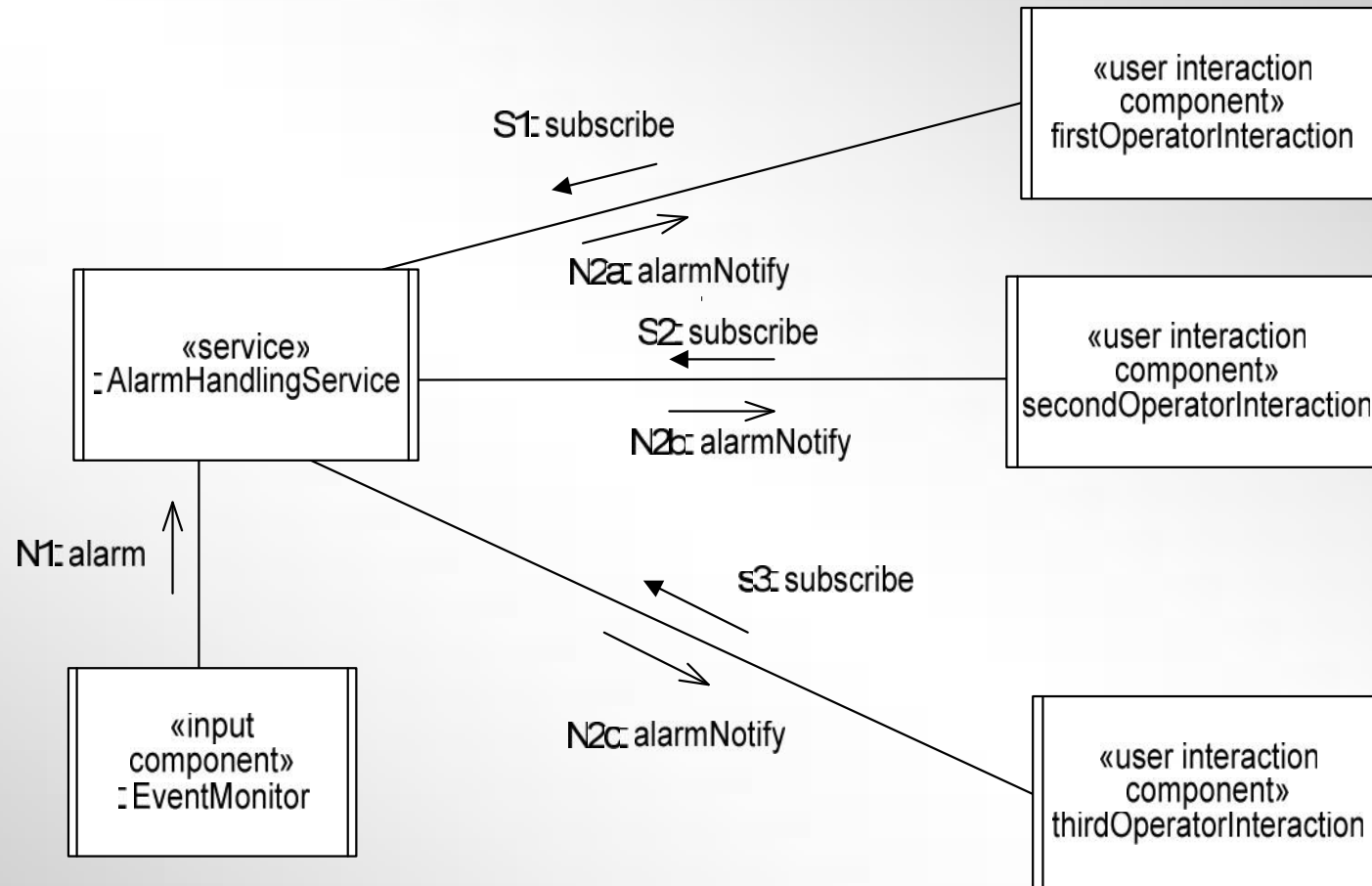


# Architectural Communication Patterns

- Asynchronous communication patterns
- Synchronous communication patterns
- Broker Communication Patterns
  - Broker forwarding
  - Broker handle
  - Discovery
- Group Communication Patterns
  - Broadcast
  - Subscription/notification
- Broker and group communication patterns
  - Facilitate software evolution and adaptation



# Subscription/Notification Pattern

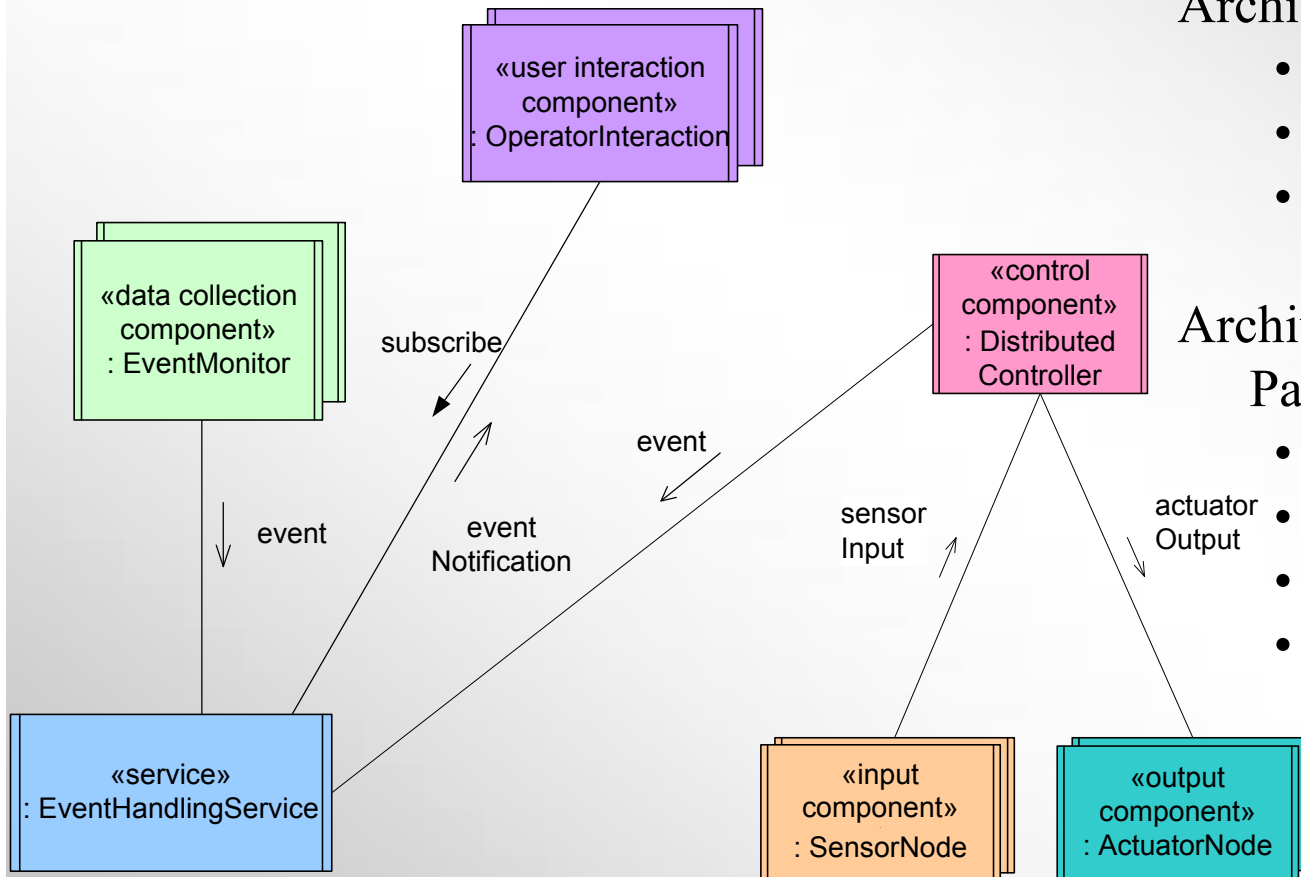


- Subscription/Notification Pattern
  - Client subscribes to join group
  - Receives messages sent to all members of group

# Building Software Applications from Software Architectural Patterns

- Consider architectural structure patterns
  - Different patterns can be combined
- Start with layers of abstractions pattern
  - Incorporate client/service patterns
  - Incorporate control patterns
- Apply architectural communication patterns
  - Decouple sender components from receiver components
    - Broker patterns
    - Group communication patterns

# Building Emergency Monitoring System From Software Architectural Patterns



## Architectural Structure Patterns

- Layered pattern
- Client/Service pattern
- Distributed Control

## Architectural Communication Patterns

- Synchronous
- Asynchronous
- Broker
- Subscription/Notification

# Conclusions

- Software architectural patterns help with
  - Designing and implementing application software architecture
  - Evolution and/or dynamic adaptation of software architecture and implementation



# Automatic extraction of (musical) metadata from audio signals: Successes, failures and challenges

Wolfgang Fohl

Dept. Computer Science  
Hamburg University of Applied Sciences

Steinberg Media Technologies

Panel: Looking for Patterns in Content: From Design to  
End-Users Consumption  
CONTENT11 2011-09-28, Rome, Italy

## Applications and Uses

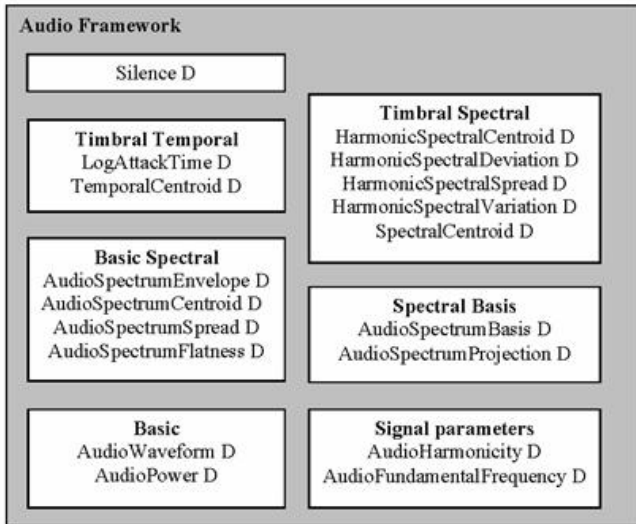
- Advanced music search
- Music production / mix / editing
- Generate playlists / DJ
- Recommend music
- Detect plagiarism / illegal downloads

# Metadata Examples

## Automatically Extractable Metadata?

- Speech ✓ / Music ⚠ / Other
- Text ⚠
- Melody
  - Single voice ✓
  - Multiple voices ⚠
- Rhythm
  - bpm ✓ – ⚠ depending on genre
- Musical Genre ✓ / ⚠
- Instrument family ✓, type ✓, individual instrument ⚠
- Player ? / interpretation ?
- “Stradivariness” of a violin ??

# MPEG-7: Audio Framework



# When Automatic Extraction Fails

## Inappropriate **Context** Information

- Proper information of musical context required
- Adequate size of **Temporal** context
- Not all context information present in signal
  - Automatic music transcription ❌
  - Contemporary music ❌

## Art: Destroy and Create Contexts

Uncertainty about context:

- Essential element of contemporary art (at any time)

Example: JOHN CAGE



Reutlingen  
University

PATTERNS 2011, September 29, 2011 – Rome, Italy

Panel discussion

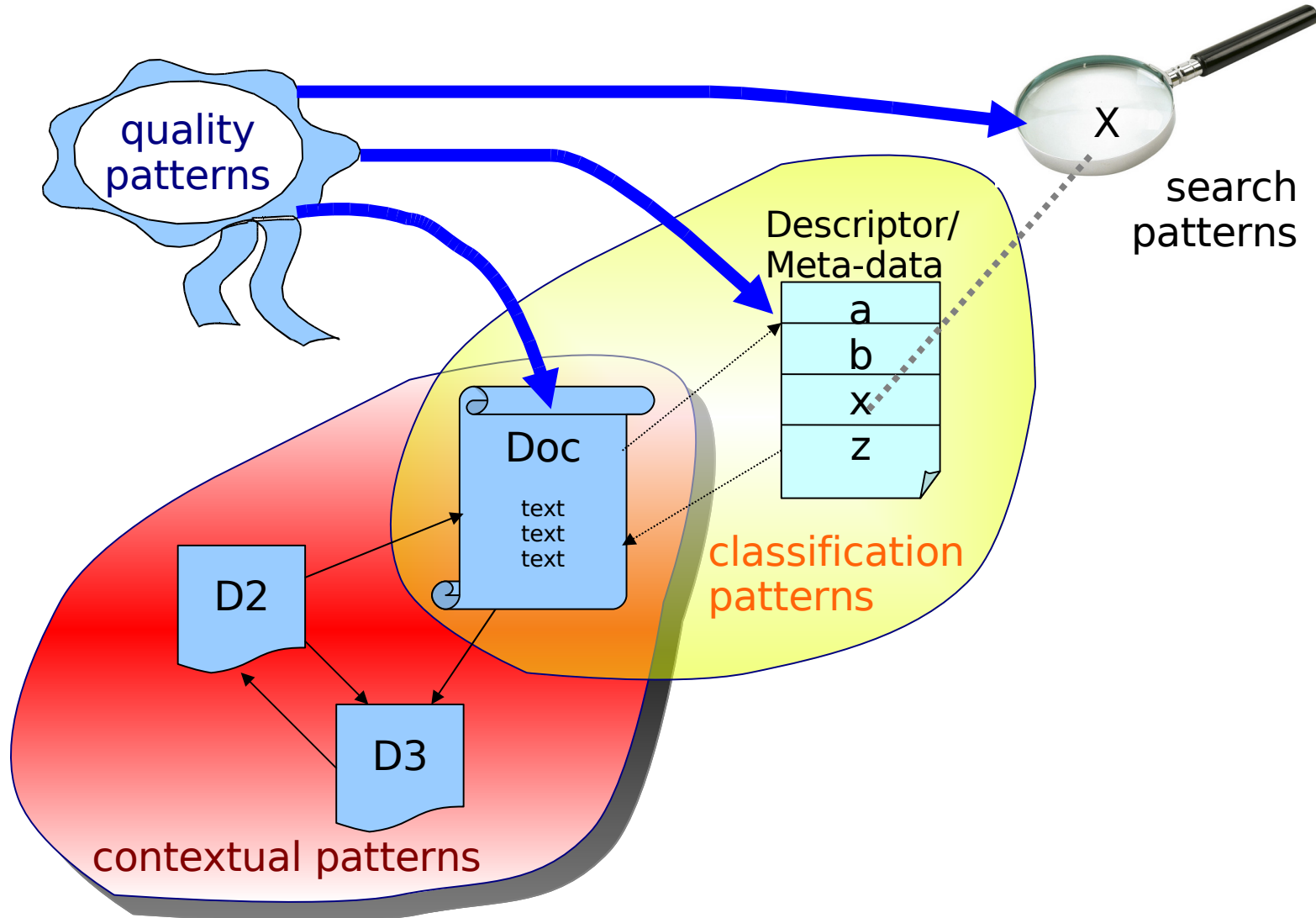
# Looking for Patterns in Content

Fritz Laux  
Reutlingen University





## Patterns Overview





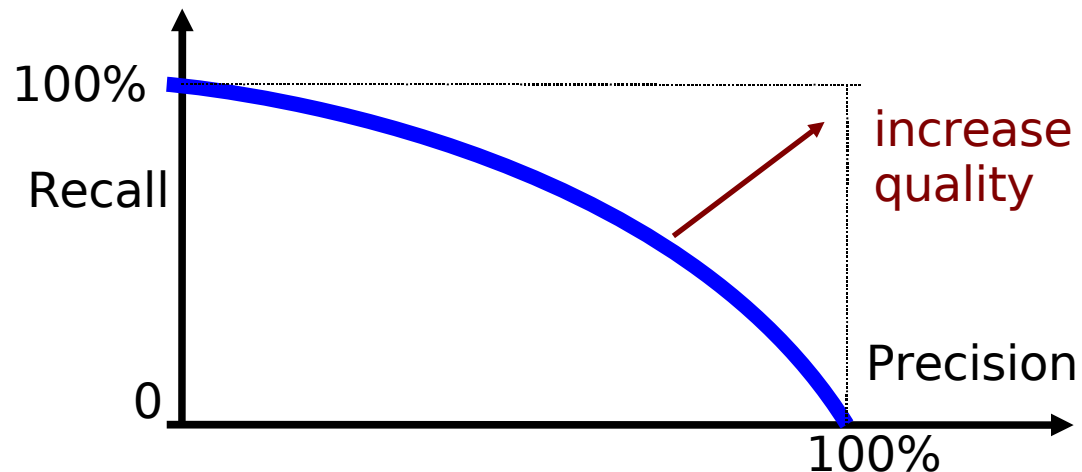
## Search Quality Patterns

Given text/document collection

How to ensure high recall and precision?

Solution

- use ontology/thesaurus
- refine/broaden search
- check actuality
- rank results
- follow and analyse links (in case of web docs)







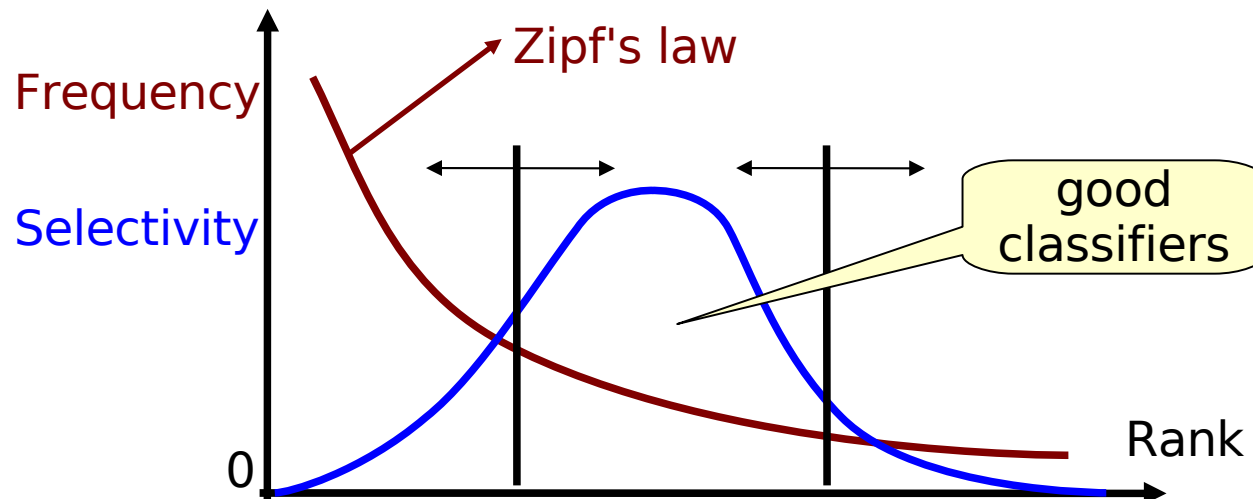
## Classification Quality Patterns

Given automated descriptor/classification

How to ensure good descriptors/correct classification?

Solution

- Descriptor derived from Structure (Title, Keywords, related work, bibliography)
- Descriptor derived from content (word frequency)
- use bibliographic Meta-data
  - use Thesaurus, Ontology
  - do sequential pattern analysis





## Document Quality Patterns

Given text/document collection

How to assess document quality?

Criteria

- understandability, consistency
- actuality, accuracy
- structure
- author's expertise/reputation

