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.
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.

- The ASIP has four phases:

  **Phase 1**
  - Sample acquisition

  **Phase 2**
  - Schema inference
  - Feedback questions
  - Unhappy with schema:
    - Modify samples
    - Modify schema

  **Phase 3**
  - Prototype generation

  **Phase 4**
  - System generation
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.

<table>
<thead>
<tr>
<th>Language-independent Content</th>
<th>ContentStructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Language Content</td>
<td>enText : String</td>
</tr>
<tr>
<td>en/US Content</td>
<td>deText : String</td>
</tr>
<tr>
<td>de/DE Content</td>
<td>itText : String</td>
</tr>
<tr>
<td></td>
<td>jpText : String</td>
</tr>
<tr>
<td></td>
<td>numberFormat : TextFormat</td>
</tr>
</tbody>
</table>

- 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:**

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

**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
hgomaag@gmail.com

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
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

1: sendATMTTransaction (in: transaction, out: response)

- «client» : ATMClient
- «service» : BankingService
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

- 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

Copyright 2011 H. Gomaa
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
Content = Signal + Metadata

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 ❓
- “Stradivarianess” of a violin ❓ ❓
MPEG-7: Audio Framework

Audio Framework

- Silence D
- Timbral Temporal
  - LogAttackTime D
  - TemporalCentroid D
- Basic Spectral
  - AudioSpectrumEnvelope D
  - AudioSpectrumCentroid D
  - AudioSpectrumSpread D
  - AudioSpectrumFlatness D
- Timbral Spectral
  - HarmonicSpectralCentroid D
  - HarmonicSpectralDeviation D
  - HarmonicSpectralSpread D
  - HarmonicSpectralVariation D
  - SpectralCentroid D
- Spectral Basis
  - AudioSpectrumBasis D
  - AudioSpectrumProjection D
- Basic
  - AudioWaveform D
  - AudioPower D
- Signal parameters
  - AudioHarmonicity D
  - AudioFundamentalFrequency D
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**
Panel discussion

Looking for Patterns in Content

Fritz Laux
Reutlingen University
Panel Discussion: Looking for Patterns in Content

Patterns Overview

quality patterns

search patterns

descriptor/meta-data

classification patterns

contextual patterns

D2

D3

Doc
text
text
text

X

a

b

x

z
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)

![Graph showing recall and precision relationship](image)

- Increase quality by adjusting recall and precision trade-off.
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

![Graph showing Frequency vs. Selectivity with Zipf's law and good classifiers highlighted.]

Reutlingen University
Document Quality Patterns

Given text/document collection

How to assess document quality?

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

in literature: $Q(\#2) > Q(\#1)$,

and in science?