

Content / Patterns Panel

Discovering Content via Patterns

*Hans-Werner Sehring, T-Systems Multimedia Solutions GmbH, Germany
(Moderator)*

Alfred Zimmermann, Reutlingen University, Germany

Samuel Kosolapov, Braude Academic College of Engineering, Israel

*Shou Boon (Schubert) Foo, Center of Social Media Innovations for Communities /
Nanyang Technological University, Singapore*

Discussion

The Patterns conference favors a comprehensive notion of *pattern*. Therefore, the predominant question raised was that for the kind patterns to be considered. In particular, patterns can serve as blueprints for particular kinds of assets or as characteristics for the identification of specific content properties.

A question from the audience was asked for the kind of abstraction that patterns provide above basic modeling paradigms like, e.g. object-orientation. The panel agreed that patterns do not add to expressiveness, but instead provide best practices, enhance the recognition of structures, and serve documentation purposes.

A discussion on abstractions emerged. It raised the question of the purpose and value of abstractions. Despite the agreed-upon value of abstractions in computer science the applicability for content was discussed controversially. When reasoning on the basis of abstractions one needs to be able to return to concrete facts. The point was made that this is not possible for content in a unique manner.

Conclusions

Conclusion 1 (non-agreed)

Information – more precisely content representing information – abstracts from phenomena and thus condenses information. The information “lost” needs to be recovered, though maybe in a non-formal way. Patterns may be used to this end.

Conclusion 2

Patterns provide indications on evidence that needs to be supported by examples (they question: “why”). Such examples are provided by content. This conclusion shows the link between content and patterns.

Conclusion 3

Languages, and thus content, are separated into natural and artificial languages. Patterns are defined w.r.t. a language. Therefore, there have to be corresponding kinds of patterns like, e.g., “computable” and “social” patterns. Since languages and expressions change, also patterns depend – like content – on both context and time.



NANYANG
TECHNOLOGICAL
UNIVERSITY

CONTENT 2012

Sentiment Analysis for Discovering Patterns in Social Media Data

Schubert Foo (sfoo@ntu.edu.sg)

**Digital Intelligence Research Cluster, Wee Kim Wee School of
Communication and Information**

Centre for Social Media Innovations for Communities (COSMIC)

Nanyang Technology University, Singapore

Introduction

- Social media growth (weblogs, discussion boards, user and critic review web sites, twitter, etc.) has fuelled research in Sentiment Analysis (SA) and Opinion Mining
- SA involves subjectivity analysis of sentiment in textual unit to determine ***sentiment polarities (i.e. positive or negative)*** of the opinions toward various aspects of a subject.
- Current Approaches:
 - Machine learning approach(Latent Semantic Analysis (LSA), Support Vector Machine (SVM), etc.)
 - Rule based heuristics and other linguistics approach
- Current Focus
 - Improvement in results through enhanced techniques
 - Visualization of sentiment summarization results
- Example Domains
 - Movie reviews - Health forums

Rule based heuristics linguistics approach

- Uses grammatical dependencies and prior sentiment scores of words in sentences.
- Requires general (e.g. *SentiWordNet*) and domain specific lexicon.
- Performs the **clause-level** sentiment analysis.
- Predict clause **sentiment score** (between negative -1 and positive 1)
E.g. “*This movie is superb.*” (+0.875)
 > “*This movie is good.*” (+0.536)
- Determines an **aspect** of a target subject (i.e. a movie) for each clause.
 - E.g., “*I love the story but I hate the music.*”
 - *I love the story.* (+0.375, *story*)
 - *I hate the music.* (-0.625, *music*)

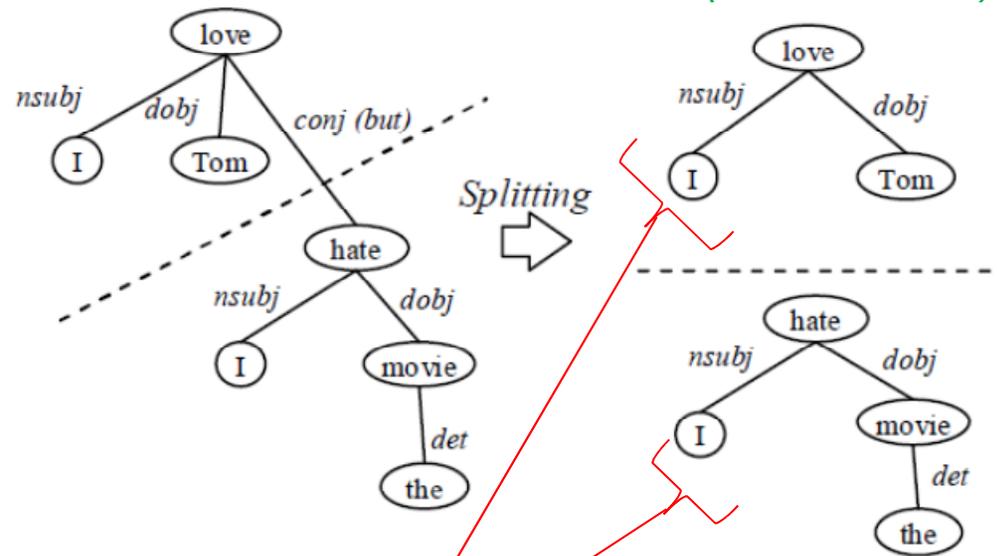
```
FOR each sentence
  Perform Semantic Annotation (such as
  movie and director names)
  Assign prior sentiment scores to words
  Generate grammatical dependencies
  Break a sentence into clauses
  FOR each clause
    Calculate clause-level Sentiment Score
    Subject or Object (Adjective + Noun)
    Verb phrase (Adverb + Verb)
    Predicate (Verb phrase +
    Object/Complement)
    Clause (Subject and Predicate)
    Determine Aspect
  END FOR
  Calculate sentence-level Sentiment Score
END FOR
```

Rule based heuristics linguistics approach

- Uses *Stanford Parser* to build a dependency tree.
- Calculates contextual sentiment scores by traversing the dependency tree.

"I love Tom but I hate the movie."

nsubj[love-2, I-1]
 dobj[love-2, Tom-3]
 nsubj[hate-6, I-5]
 conj_but[love-2, hate-6]
 det[movie-8, the-7]
 dobj[hate-6, movie-8]



(+0.375, cast)

(-0.625, overall)

Pos/Neu **Verb** and Pos/Neu **Object** => $(|O| + (1-|O|) * |V|)$
 E.g., "love Tom": +0.375 and +0.0 => $(0 + (1-0.0) * 0.375) = +0.375$

Neg **Verb** and Pos/Neu **Object** => $-|V|$
 E.g., "hate movie": -0.625 and +0.0 => -0.625

User Interface (Example: Avatar movie)

MovieRev-Fed
The trends are here ...

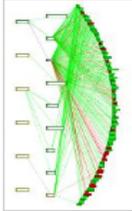
HOME MOVIES DIRECTORS CASTS SEARCH ABOUT US



Title - Avatar
Directed By - [James Cameron](#),
Casted By - [Sam Worthington](#), [Zoe Saldana](#), [Sigourney Weaver](#),
[Stephen Lang](#), [Joel Moore](#), [More Details](#)

Review Type: Critic Reviews
Review Section: 0

[Show Full Review](#)



SentiGraph



Sentiment Term Cloud

best	great	wise	bad	wrong	hell
perfect	wisely	amazing			hate
Positive			Negative		
gruffly			fear	not	puny
			worst	shock	
			sheer		

Sentiment Treemap

Sentiment Score

Total Number of reviews 5 (C:5,B:0,D:0,U:0,T:0)

Positive/Negative Reviews: neg 40% 60% pos

Sentiment Score: 0.0480

Aspects

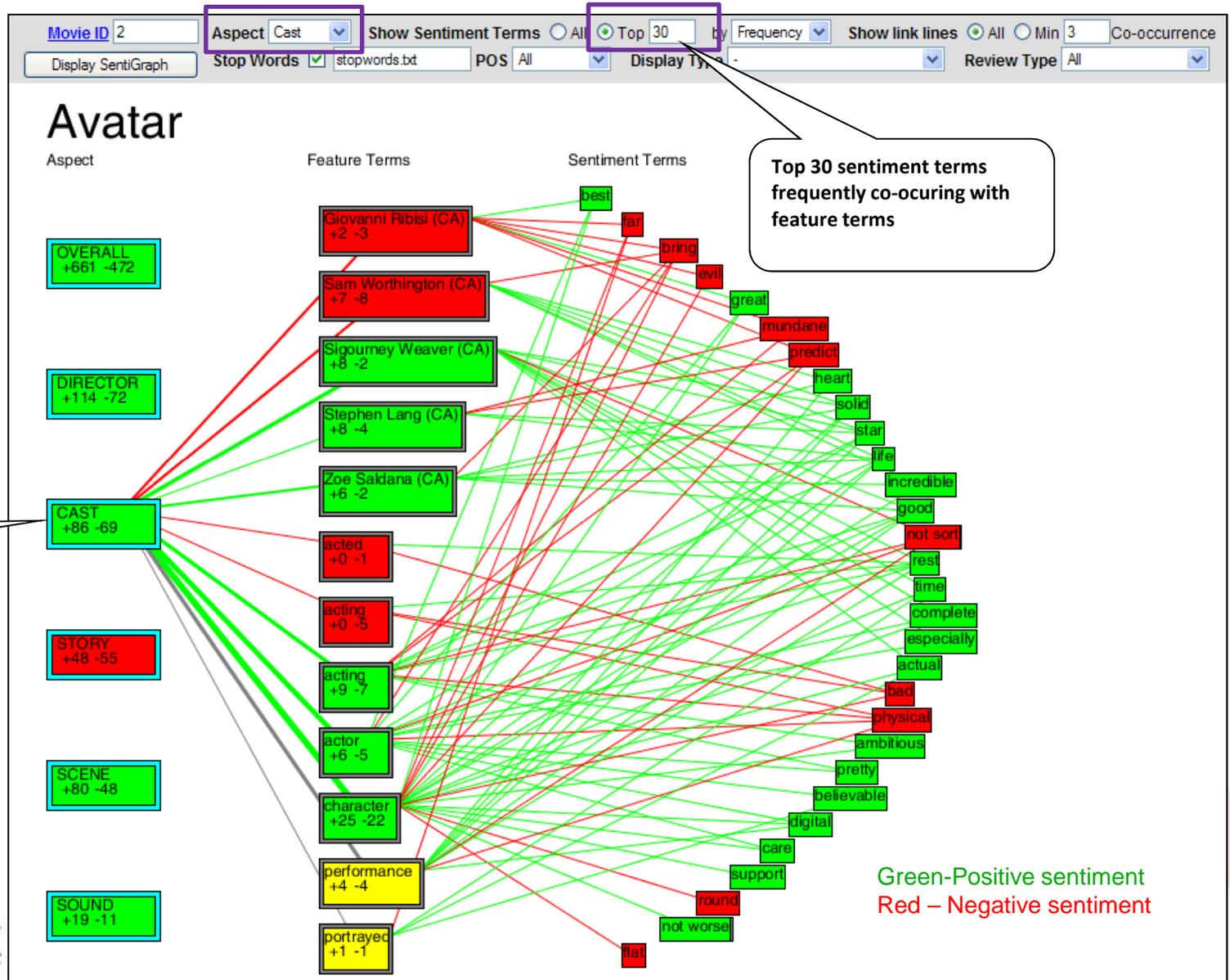
Review Aspect	Number of Clauses	Positive/Negative Posts	Sentiment Score
Overall Aspect	186	neg 40% 60% pos	0.0632

Aspect-based Sentiment Summarization

Sentiment Bar

Sentiment Meter

SentiGraph - Visualization Parameters



Challenges in Sentiment Analysis

- Subjectivity – sentiments prone to human disagreement
- How to address culture and context factors, and linguistics nuances?
- Language “quality” dependent. Multi-lingual SA
- Most SA algorithms uses simple terms – poor accuracy?
- Short text and jargons (e.g. twitter feeds) don’t help.
- Visualization of sentiment analysis results.

References:

Na, J.C., Thet, T.T, Khoo, C.S.G, & Kyaw, W.Y.M. (2011). Visual Sentiment Summarization of Movie Reviews. Proc. *13th International Conference of Asia-Pacific Digital Libraries (ICADL2011)*, Beijing, China, October 24-27.

Na, J.C, Kyaw, W.Y.M., Khoo, C.S.G., Foo, S., Chang, Y.K., & Theng, Y.L. (2012). Sentiment Classification of Drug Reviews Using a Rule-Based Linguistic Approach. Accepted for presentation at *14th International Conference of Asia-Pacific Digital Libraries (ICADL2012)*, Taipei, Taiwan, November 12-15.

Thank You

Schubert Foo (sfoo@ntu.edu.sg)



Getting Instant Feedback from multi-camera Video Stream

Samuel Kosolapov

Goal: Getting Instant Feedback
from students
(or, say, participants of this panel)
by real-time search for specific patterns
in multi-camera video stream



Items to discuss:

Realization Options

Will it work in the real-life?

Instant Feedback System (IFS) in Education.

Pedagogic Goals

To provide instant grading (micro-exams)

To improve learning and teaching
(by analyzing students answers)



Are they with me ?

Am I clear enough ?

Which part(s) of my lecture
need improvement(s) ?

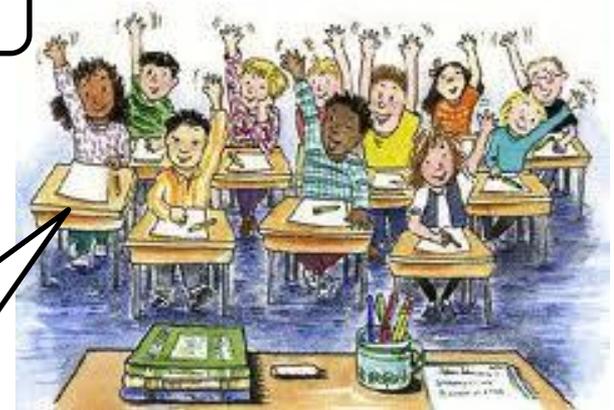
Instant Feedback System (IFS)

Technology: Clickers

versus multi-camera stream analysis



IR Clickers



Real-time search for specific patterns
in multi-camera video stream
Implementation options ?
Will it work in the real life ?

When asked,
Every student raises
his "own" simple "object"
(passive, no batteries)
Image processing software
provides instant grading

Discovering Content via Patterns.

Panel discussion, CONTENT 2012, Nice, France.

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 structures.
 - For content analysis / schema derivation.
 - For content recognition and content “type checking”.
 - For content representation.



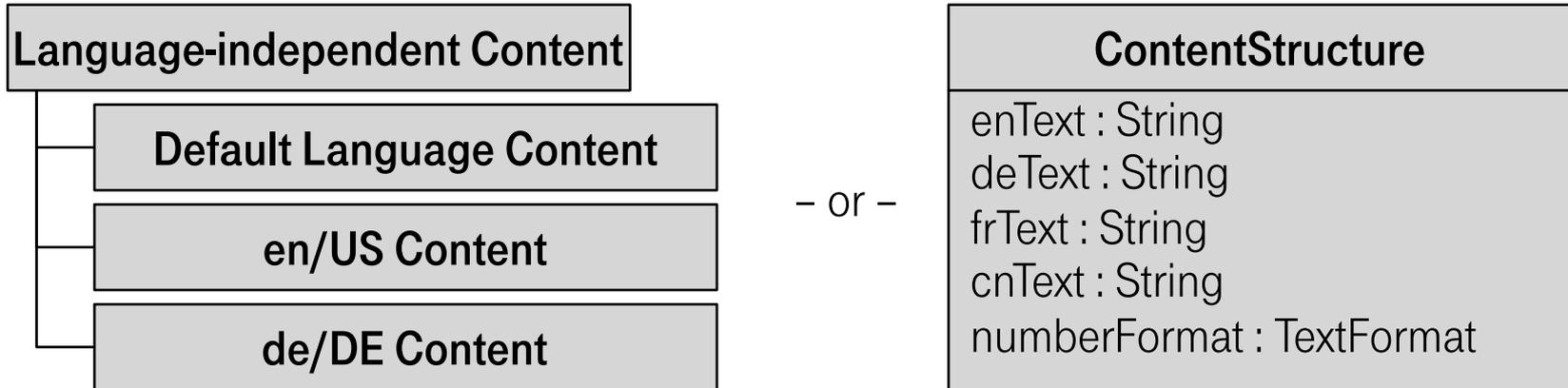
On the Notion of Pattern.

- **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 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 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.

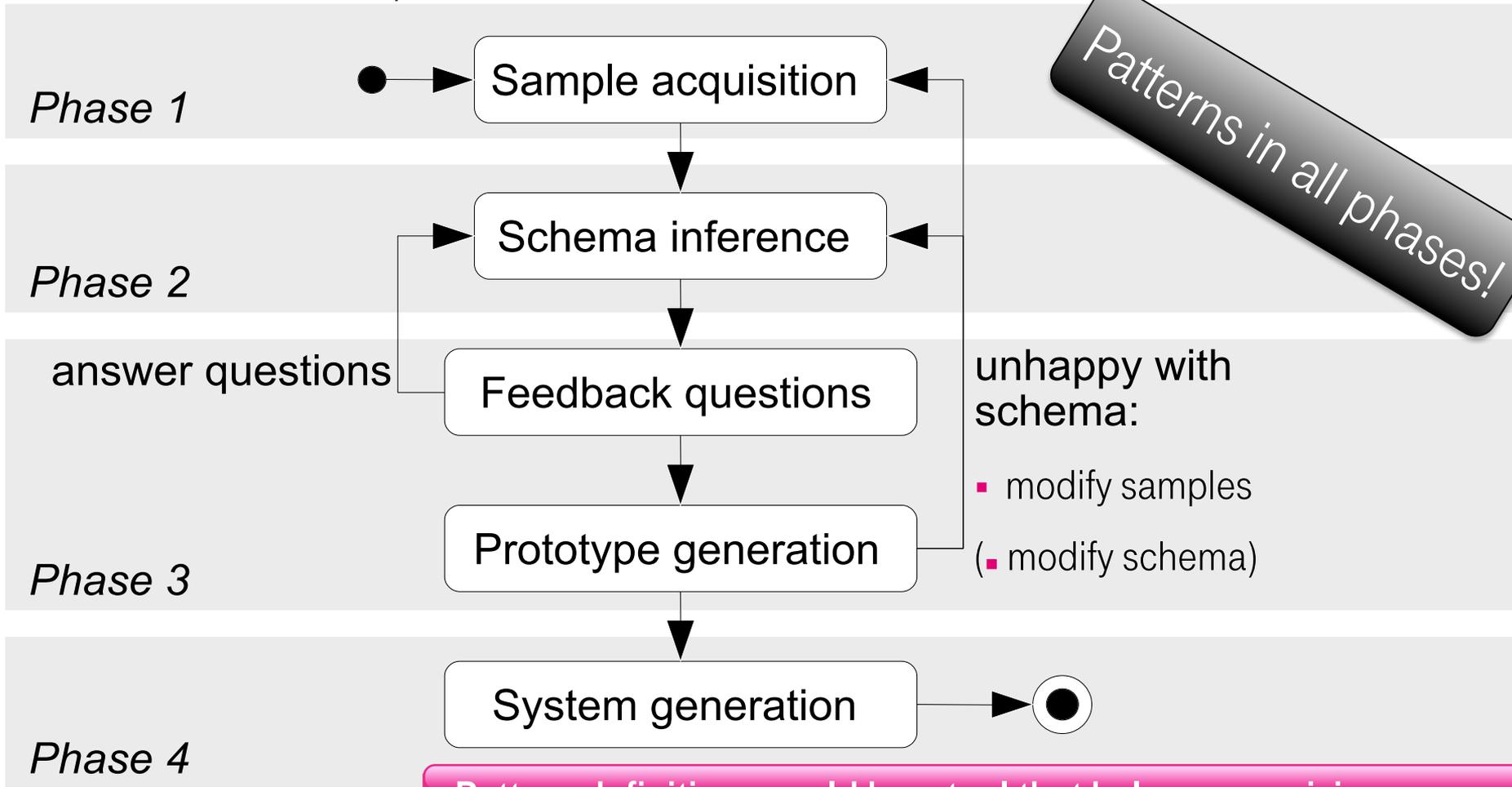
Pattern definitions could be a tool that helps designing content structures that support typical problems in a more adequate way.



Patterns for Content Analysis.

Example: The Asset Schema Inference Process (ASIP).

- The ASIP has four phases:



Patterns in all phases!

answer questions

unhappy with schema:

- modify samples
- (▪ modify schema)

Pattern definitions could be a tool that helps recognizing content structures that are typical for certain content instances.



Patterns for Content Representation.

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

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



Typical approach: Reduction to lower level.



Thank you.
Let's discuss.



Pattern Discovery Panel: Software Patterns

Alfred Zimmermann

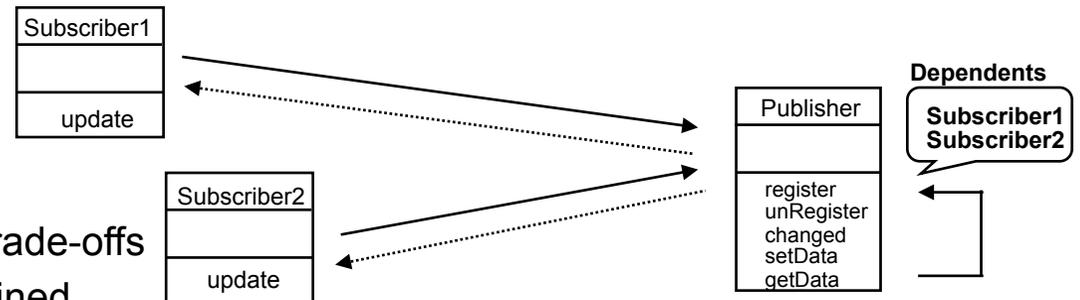
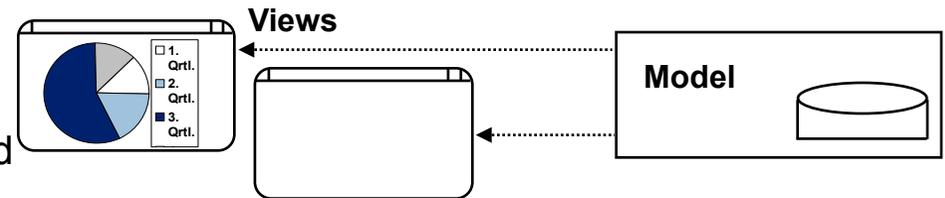
Reutlingen University, Germany

Software Patterns

Basic Concepts

- **Pattern**

- **Core** of a **solution** for a **recurring problem** that occur in a **specific context**
- **Supports humans** (and not machines), based on **verified design** and **practical experience**
- Articulates a **common vocabulary** of concepts and knowledge
- **Essential elements:** extending the causal problem – solution pair
 - **Pattern Name**
 - **Problem**
 - **Solution**
 - **Consequences** – as results and trade-offs



- **Pattern Catalog:** a collection of self-contained patterns, indexed by categories
- **Pattern Encyclopedia:** an index of patterns
- **Pattern System:** a collection of patterns with references between related patterns
- **Pattern Language:** a network (graph) of tightly connected patterns giving solution chains for specific problems

Software Patterns

Patterns Writer Workshop

1. The paper is discussed by a **group of people** including its **author**, a **moderator**, and a group of **reviewers** who are familiar with the contents of the paper.
2. The **author of a paper reads a paragraph** of his/her choice, pointing to what is particularly important about the pattern.
3. One or two **reviewers briefly summarize the paper**. The goal is to identify what the reviewer(s) thought were the **key points** of the pattern.
4. The **group** then **discusses** what they liked about the paper, first in terms of content and then in terms of style.
5. After presenting the **positive aspects** of the paper, the group discusses **how to improve** the content and style of the paper. The goal here is *not* to criticize the paper per se, but rather to give the author constructive suggestions on how to make the paper better.
6. After this discussion, the **author** of the paper may **ask questions** of the **reviewers** to clarify their statements. The goal is to give the author a chance to better understand certain comments, rather than to defend the paper.
7. The session closes with the **audience thanking the author** for writing the paper.

Pattern Innovation and Evolution Process

Pattern Life-Cycle Status Information

- The **Pattern Innovation and Evolution Process** controls the **pattern's state**, which is used to **track the development** of a pattern over time
- **Associated pattern status information:**
 - **Just created:** patterns were recently submitted as a non-validated idea
 - **Under consideration:** the pattern looks promising, but needs further evaluation
 - **Pattern candidate:** states, that the pattern is close to being approved
 - **Approved:** finalizes the pattern review process and settles the pattern

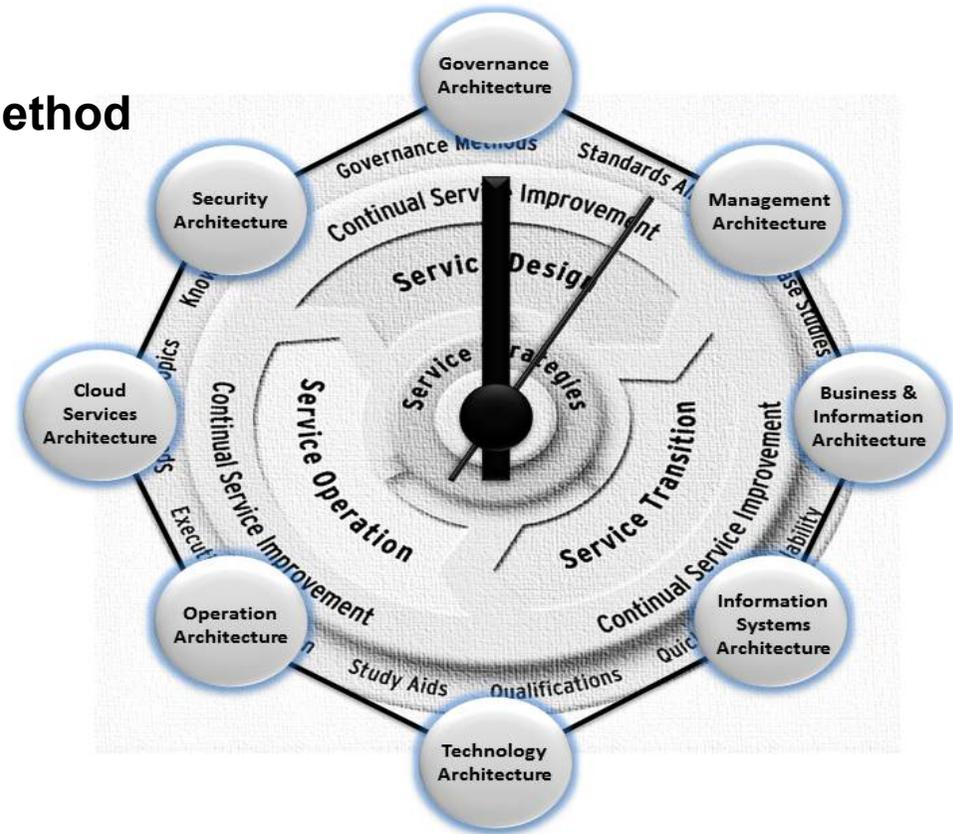
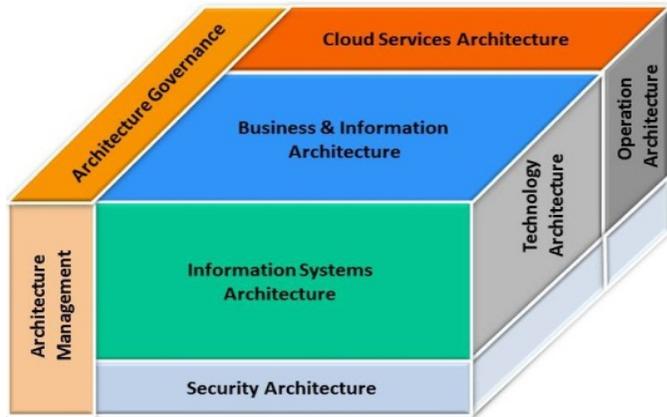
Pattern Innovation and Evolution Process

Pattern Origin

- In addition patterns can **originate** both from the **project** itself and from **external sources**
- **Pattern Origins:**
 - **Derived from project:** The pattern derives directly from the work within the project and will automatically be assigned the state under consideration. The pattern will be reviewed, perhaps re-worked and finally validated through an approval process by a validator
 - **Adapted to project:** The pattern originates from external sources, but has been adapted for use in the context of the project
 - **External:** The pattern exists in other pattern collections (e.g. a standard UI pattern) and is implemented in the current project's products and services

PEAM – Pattern-oriented Architecture Development Method

- **PEAM – Pattern-oriented Enterprise Architecture Development Method** supports a navigable process model for iteration through all eight **ESARC Reference Architectures**
- The **PEAM Architecture Development Method** relies on the idea of **software patterns**
- The Architecture Development Method is based on **ESARC, TOGAF-ADM** and the **ITIL service-oriented lifecycle**

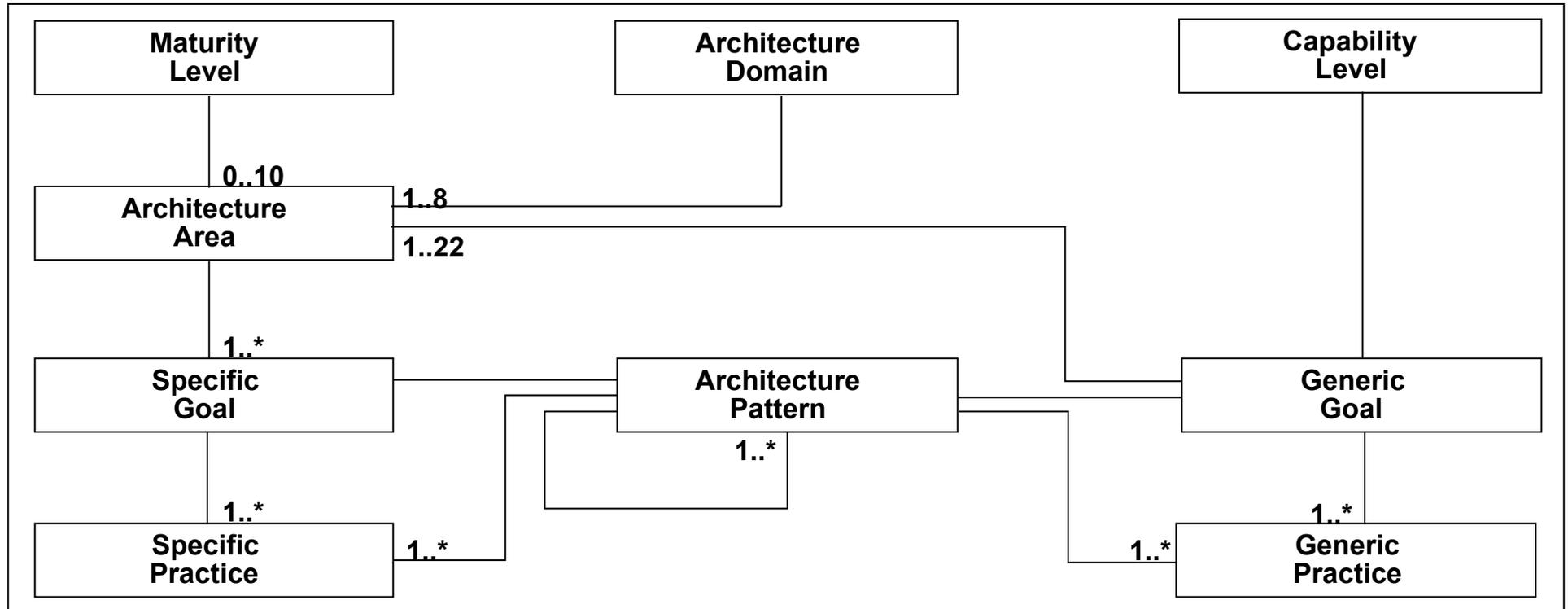


SOAMMI (SOA Maturity Model Integration)

Metamodel (adapted from TOGAF, CMMI, a. o.)

Hypothesis of our research

- CMMI (Capability Maturity Model Integration) is well known as suitable framework to assess software processes, but not software architectures, nevertheless the meta model of CMMI can be extended to evaluate architecture capabilities of service-oriented systems
- The idea of a pattern language can be applied consistently for both architecture assessments and optimization, setting a base for effective cyclic evaluations of service-oriented systems



SOAMMI Business Architecture

Architecture Area: BPR Business Processes & Rules

■ Purpose

Structure, design, model, and represent business value chains and business processes to support modeled products and services

■ Maturity Level: 2

■ SG 1: Model Business Value Chains as Root of Business Processes

- SP 1.1 Identify business value for business operations
- SP 1.2 Structure value chains
- SP 1.3 Optimize business considering customer channels and supplier networks

■ SG 2: Model and Optimize Business Processes

- SP 2.1 Identify business activities for business processes: system activities, user interaction activities, manual activities
- SP 2.2 Structure business processes for business roles and organizational units
- SP 2.3 Define business workflows and business process rules
- SP 2.4 Model and represent business processes

■ SG 3: Model and Represent Business Control Information

- SP 3.1 Identify and represent control information for product monitoring
- SP 3.2 Identify and represent control information for process monitoring

SOAMMI Business Architecture

Business Process Pattern

■ **Problem**

How can we structure, optimize and model business processes, related workflows, and business process rules?

■ **Solution**

- Identify business activities for business processes: system activities, user interaction activities, manual activities
- Structure business processes for business roles and organizational units
- Define business workflows and business process rules
- Model and represent business processes

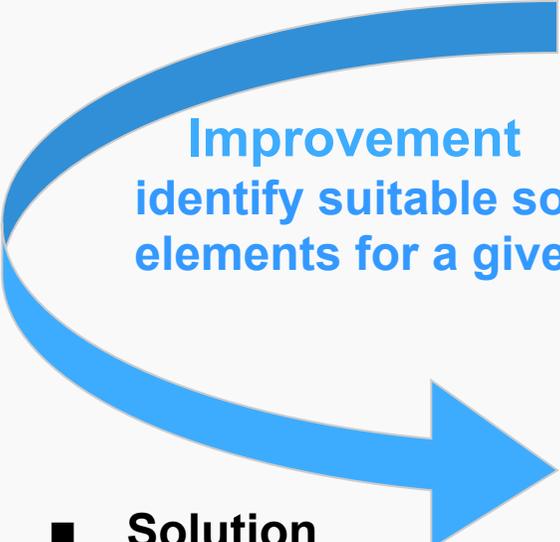
Service-oriented Architecture Capability Patterns

Architecture Diagnostics and Improvement

Software quality patterns can be applied consistently for architecture quality assessments for both capability diagnostics and for quality improvements of service-oriented architectures

- **Problem**

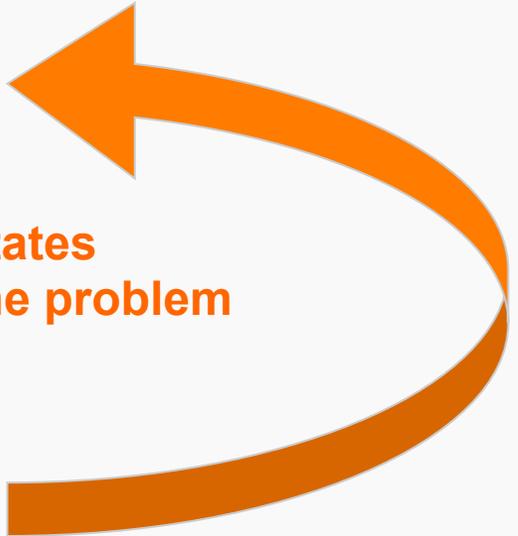
What is the problem to be solved using the pattern?



Improvement
identify suitable solution
elements for a given problem

- **Solution**

What are solution elements suitable to solve the problem?



Diagnostics
verify solution states
and determine the problem