Architecture Knowledge Management

Prabhakar TV
IIT Kanpur
tvp@iitk.ac.in

What is Architecture?

- Every entity (abstract or otherwise) has a structure
- The structure can be views at different levels of granularities, different levels of abstraction
- It will have stakeholders, who have concerns and have their specific viewpoints.
- The structure influences the properties of the artifact
- All these are related to 'architecture'
- Many architectures: Enterprise, Business, Information, Technology
- The major lesson

Quality Attributes are what drives the architecture.

Some definitions

- IEEE Standard 1471 "The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution."
- Open Group:`"Architecture has two meanings depending upon its contextual usage.

  (1) A formal description of a system, or a detailed plan of the system at component level to guide its implementation.

  (2) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time."

- **Architecture and Design**

Motivation

- What to teach?
- Searching and sharing the desk top
- Asset Management
- Handles for Designing

This talk is about

- What is knowledge?
- Knowledge representation schemes
- What is architectural knowledge?
- Representing architecture knowledge
- Some things to do

What is knowledge?

- Plato: in order to count as knowledge, a statement must be at least *justified*, *true*, and *believed*.
- Many debates in the Philosophy community
- We use the definition: *knowledge refers to a permanent structure of information stored in memory.*

Knowledge (continued)

- All activity which is not stimulus-response is knowledge based
- Knowledge (semantics) is encoded in syntax
- Computer is a syntax processor (Newell)

Let us start with

Role of Knowledge in software development, Pierre N. Robillard *CACM* '99

- **Knowledge Concepts from cognitive science**

Key Knowledge Concept	Viewpoint
Procedural/Declarative	**Knowledge nature of content**
Schema	**Knowledge of internal structure**
Proposition	**Formal Knowledge representation**
Chunking	**Representing units of knowledge**
Planning	**Managing knowledge structure**

Knowledge types(1/2)

- **Procedural**
  - Knowledge of how to do things.
  - Know-how
  - Difficult to describe in words
- **Declarative**

Knowledge types(2/2)

- **Procedural**
- **Declarative**
  - Knowledge of what is?
  - Two types: *topic* (or semantic) and *episodic*
  - Topic – meaning of words, definitions..
    - » topic knowledge is obtained from textbooks or courses or attending TOGAF workshops
  - Episodic: Memory of events that have already happened
    - » one's experience with knowledge.
    - » learned through experience once the topic knowledge is obtained
    - » Episodic memory can be used for inference and for learning procedural and declarative knowledge.

Knowledge representation

- What are we encoding – which kind of knowledge?
- How to communicate episodic knowledge?

Episodic and Topic

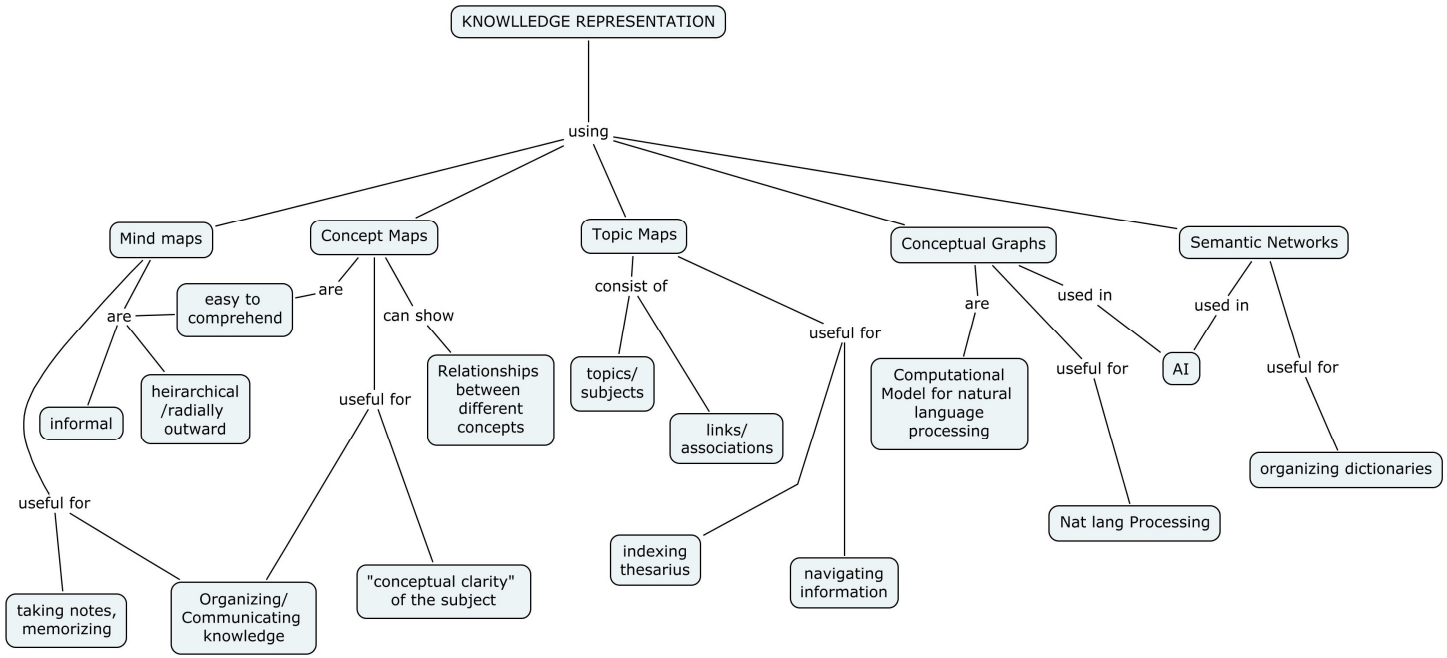
- Software development requires topic and episodic knowledge.
- Lack of episodic knowledge leads to problems
- For example: While writing code, lack of episodic knowledge in the programming language might result in a complex program.
- Novice programmers have only a limited store of episodic knowledge.
- A methodology learned from a book or a crash course is essentially based on topic knowledge.

Knowledge documentation

- knowledge structure (schema) - supports understanding
- Plans – mechanism to organize the knowledge
- We normally document the final representation of the knowledge structure, or the source code.
- But the documentation of the plan – the design rationale is new

Knowledge modelling

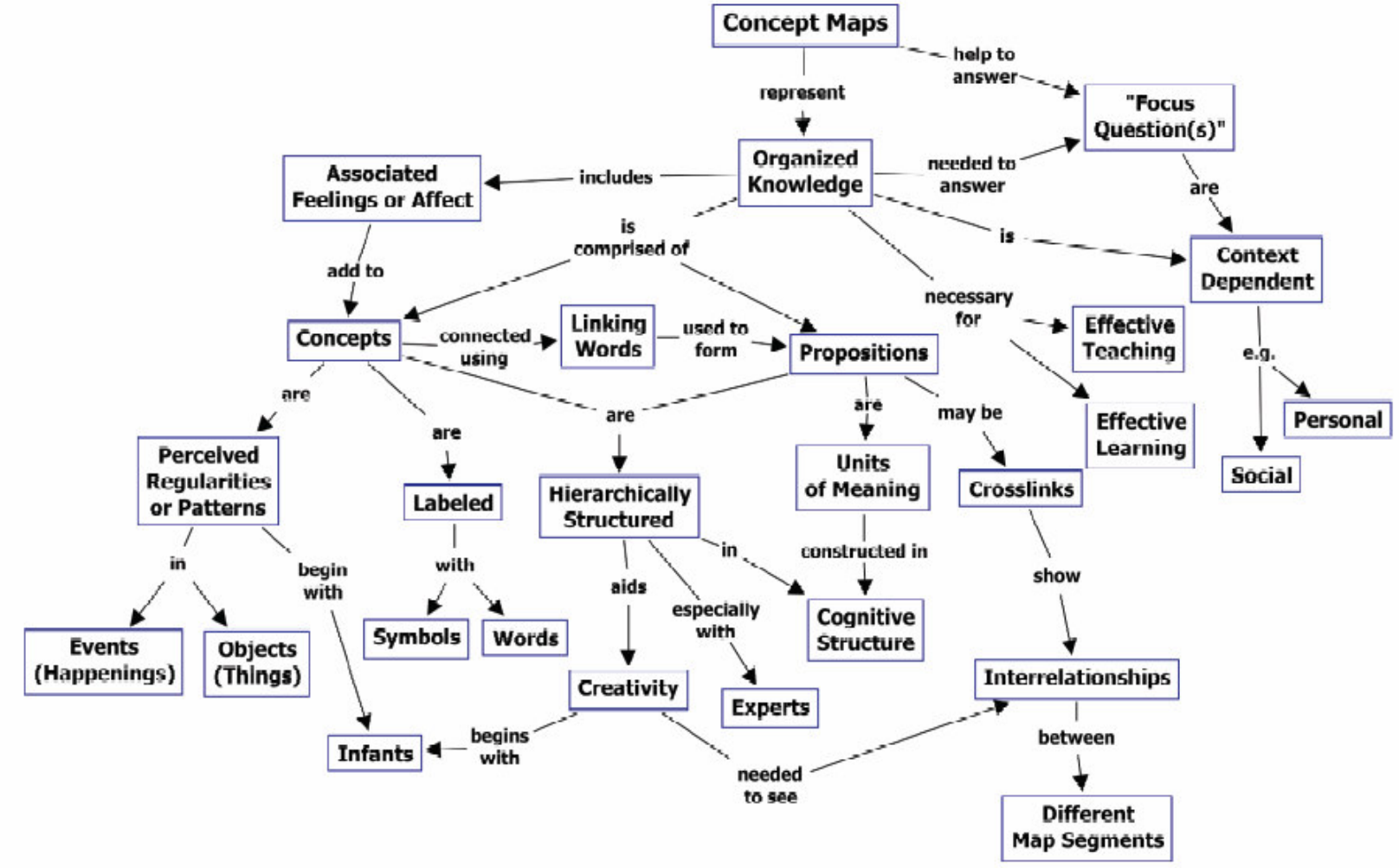
- Concept maps
- Topic maps
- Ontologies

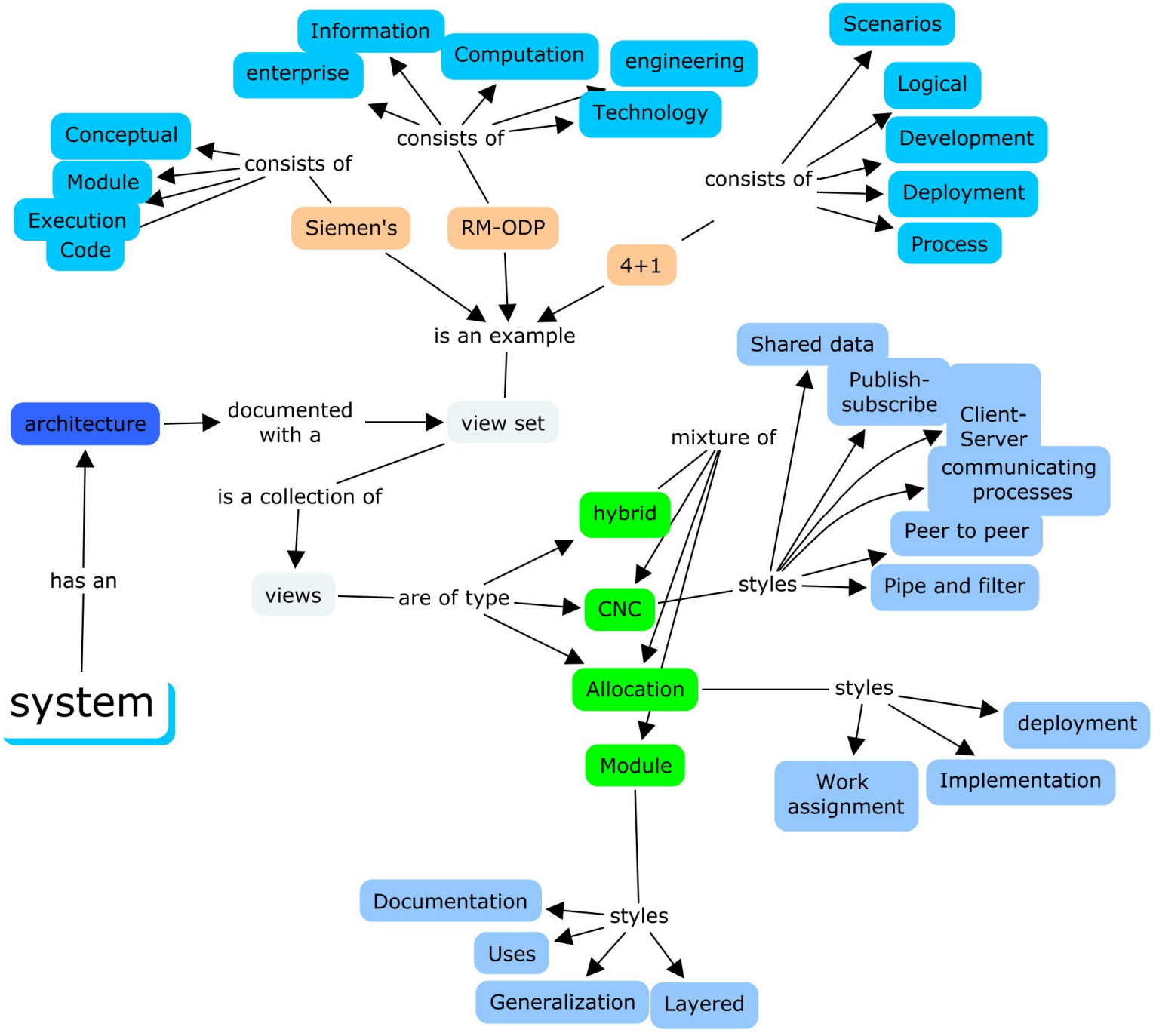


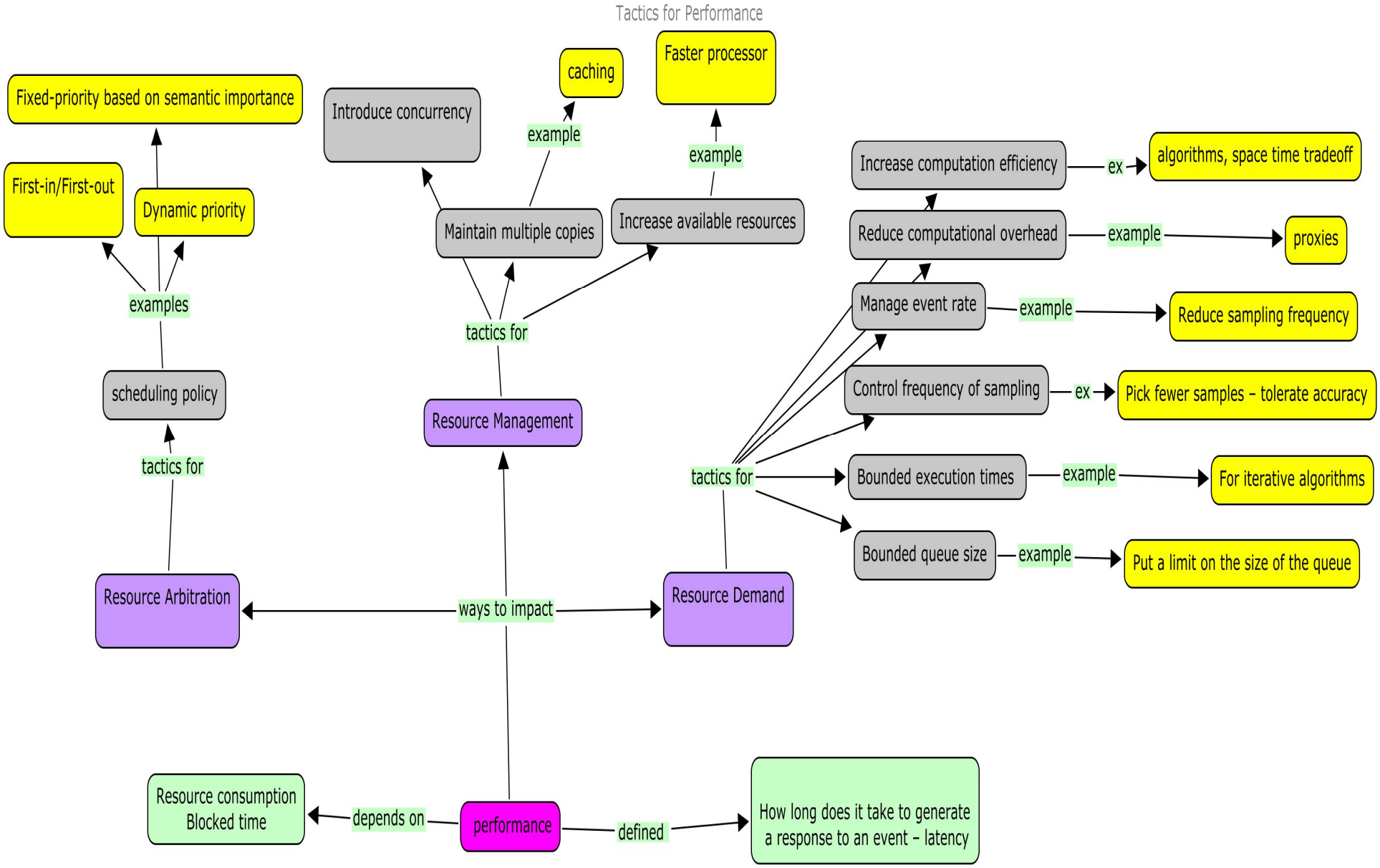
Concept maps

- **What is a concept map?**
- **Concept maps offer a method to represent information visually.**
- **Developed in the 70's by Novak at Cornell**
- **Useful for**
  - to generate ideas (brain storming, etc.)
  - to design a complex structure (long texts, hypermedia, large web sites, etc.)
  - to communicate complex ideas
  - to assess understanding or diagnose misunderstanding
  - communication and representing tacit knowledge

A Concept map on concept maps (from Novak)




Information
Computation
engineering
enterprise
Technology
consists of
Conceptual
consists of
Module
Execution
Code
Siemen's
RM-ODP
Scenarios
Logical
Development
consists of
Deployment
Process
4+1
is an example
Shared data
Publish-
subscribe
Client-
Server
architecture
documented
with a
view set
mixture of
communicating
processes
is a collection of
hybrid
Peer to peer
has an
views
are of type
CNC
styles
Pipe and filter
Allocation
styles
deployment
Module
Work
assignment
Implementation
system
Documentation
styles
Uses
Generalization
Layered


Tactics for Performance
Fixed-priority based on semantic importance
First-in/First-out
Dynamic priority
examples
scheduling policy
tactics for
Resource Arbitration
Introduce concurrency
caching
example
Faster processor
example
Maintain multiple copies
Increase available resources
tactics for
Resource Management
ways to impact
Increase computation efficiency
ex
algorithms, space time tradeoff
Reduce computational overhead
example
proxies
Manage event rate
example
Reduce sampling frequency
Control frequency of sampling
ex
Pick fewer samples – tolerate accuracy
tactics for
Bounded execution times
example
For iterative algorithms
Bounded queue size
example
Put a limit on the size of the queue
Resource Demand
Resource consumption
Blocked time
depends on
performance
defined
How long does it take to generate a response to an event – latency

Meta data to Topic maps

Metadata

- Metadata is data about data – information about objects
- Vocabulary for metadata Dublin Core
- In general "any statement about an information resource"
- Html meta tag

Some tags and their usage

- Title—Name given to a resource.
- Creator—Entity that created the resource.
- Subject—Topic of the content or resource.
- Description—Describes the content.
- Publisher—Entity that published the content.
- Date—Date associated with the resource life cycle.
- Format—The format of the resource.
- Language—Language used in the resource.

```
<head>
<title>Meta data example
    </title>
<meta NAME="DC.Creator"
    content="Mathew Bass">
<meta NAME="DC.Title"
    content="Distributed
    Software Development">
<meta NAME="DC.Date"
    content="2007-01-08">
<meta NAME="DC.Format"
    content="text/html">
<meta NAME="DC.Language"
    content="en">
<meta NAME="" content="">
</head>
```

Dublin Core Tags

1. contributor
2. coverage
3. creator
4. date
5. description
6. format
7. identifier
8. language
9. publisher
10. relation
11. rights
12. source
13. subject
14. title
15. type

What to do with it?

- **Most useful metadata – keyword**
  - It describes what the document is about

- **The other metadata are useful in managing the documents**
- **Can be used to refine search**

Subject based Classification

- Controlled vocabularies
- Taxonomies
- Thesauri
- Faceted classification
- Ontologies
- Topic Maps

Subject-based classification

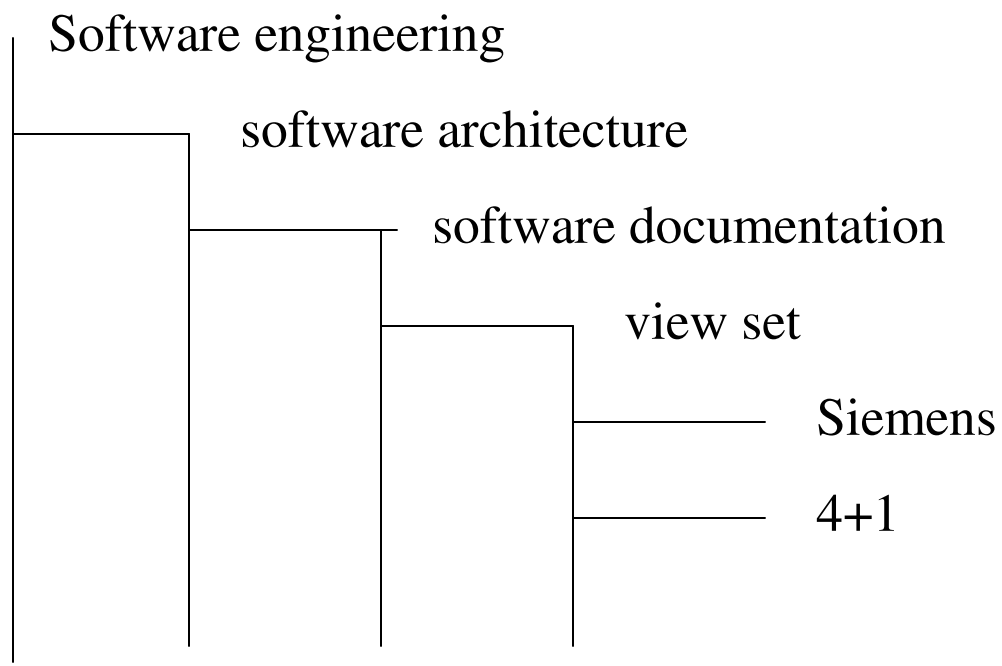
- **classification that groups objects by the subjects they are about.**
- **The metadata field related to 'what the object is' use subjects**
- **Techniques differ on what they say about the subject – not the object**
- **Metadata describes objects,**
  - by connecting objects to the subjects they are about.

Controlled vocabularies

- **a closed list of named subjects, which can be used for classification.**
- **usually known as terms,**
- **where a *term* is a particular name for a particular concept – Keyword**
- ***term* is the name of a *concept***
- **controlling vocabulary helps in**
  - eliminating meaningless terms,
  - terms which are too broad, or are too narrow,
  - preventing different authors from misspelling and choosing slightly different forms of the same term.

Taxonomies

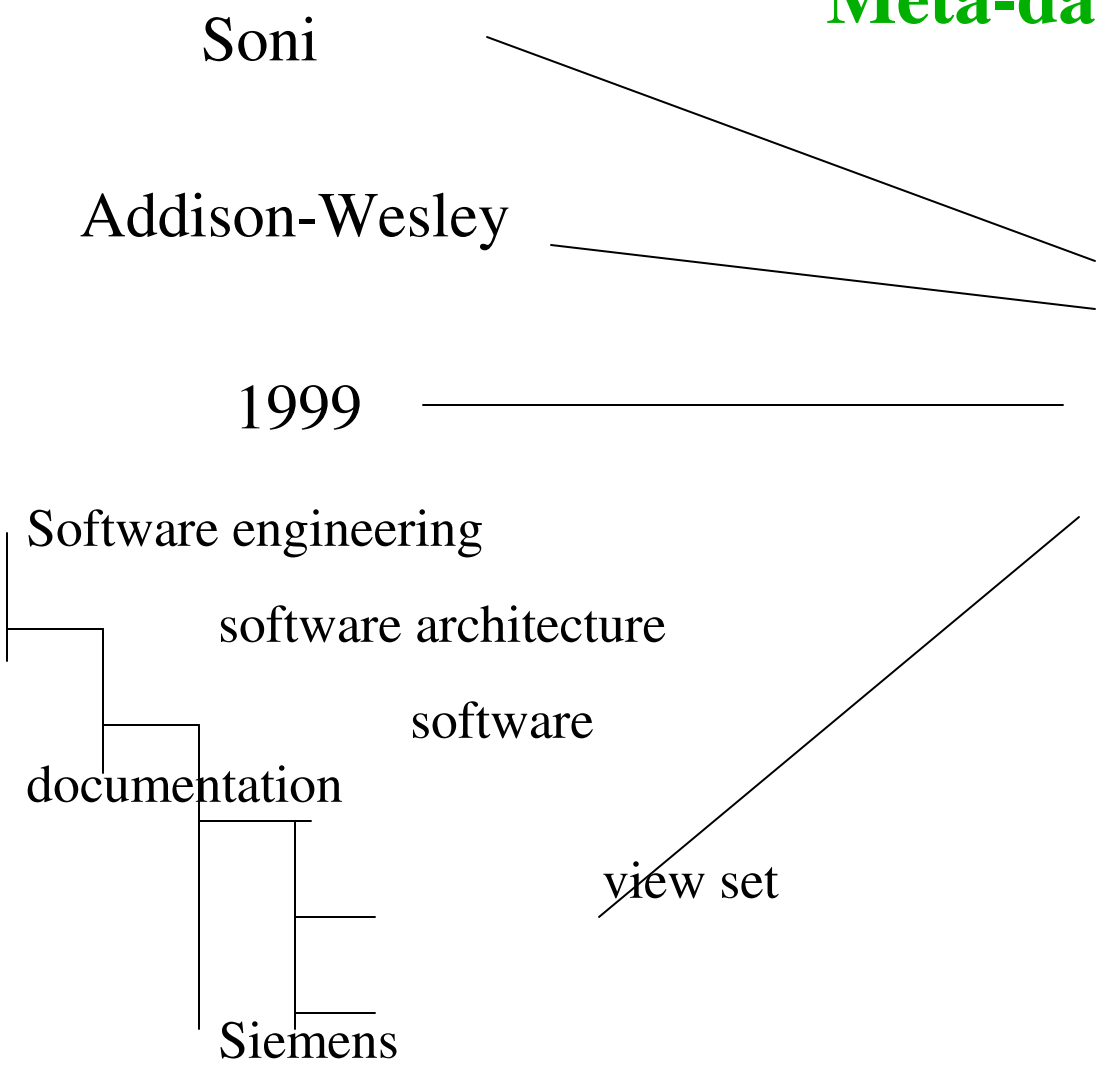
- **Most famous - Carl von Linné**
- **classification that arranges the terms in the controlled vocabulary into a hierarchy**



Taxonomy (contd.)

- helps by describing the *subjects*
- from metadata point of view - no difference between a simple controlled vocabulary and a taxonomy
- The metadata only relates objects to subjects, whereas taxonomy arranges the subjects in a hierarchy.
- A taxonomy describes the subjects being used for classification, but is not itself metadata; it can be used in metadata

Meta-data and Taxonomy



Applied Software Architecture

Hoffmeister, Nord, Soni

…..

--

--

4+1

Thesauri

- Thesauri extend Taxonomies
- Hierarchies + other statements about the subjects
- have a much richer vocabulary for describing the terms than taxonomies

Terms in a Thesaurus

- **BT**
  - "broader term", refers to the term above this one in the hierarchy;
  - inverse property known as NT
  - taxonomies are thesauri that only use the BT/NT properties.
- **SN**
  - "scope note"
  - This is a string attached to the term explaining its meaning within the thesaurus.
  - This can be useful in cases where the precise meaning of the term is not obvious from context.
- **USE**
  - Synonymous term
  - inverse property known as UF
- **TT**
  - "top term",
  - topmost ancestor of this term.
- **RT**
  - Short for "related term",
  - a term that is related to this term, without being a synonym of it or a

Faceted classification

- **Personality**
  - This facet was intended for the primary subject of the document, and is considered the main facet.
- **Matter**
  - The material or substance the document deals with.
- **Energy**
  - The processes or activities the document describes.
- **Space**
  - The locations described by the document.
- **Time**
  - The time period described by the document.

- **1930 technique from Dr Ranganathan**

Ontologies

- **Ontologies - culmination of this progression**
- **model for describing the world that consists of a set of types, properties, and relationship types**

What is *ΟΝΤΟΛΟΓΨ*?

- In philosophy,

ontology is the study of being or existence.

(from the Greek ὄν, genitive ὄντος: of being (part. of εἶναι: to be) and -λογία: science, study, theory)

What is an ontology?

Formal	machine readable
explicit specification Of a	concepts, properties, functions, axioms are explicitly defined
shared	Consensual knowledge
Conceptualization	Abstract specification of some phenomena in then world

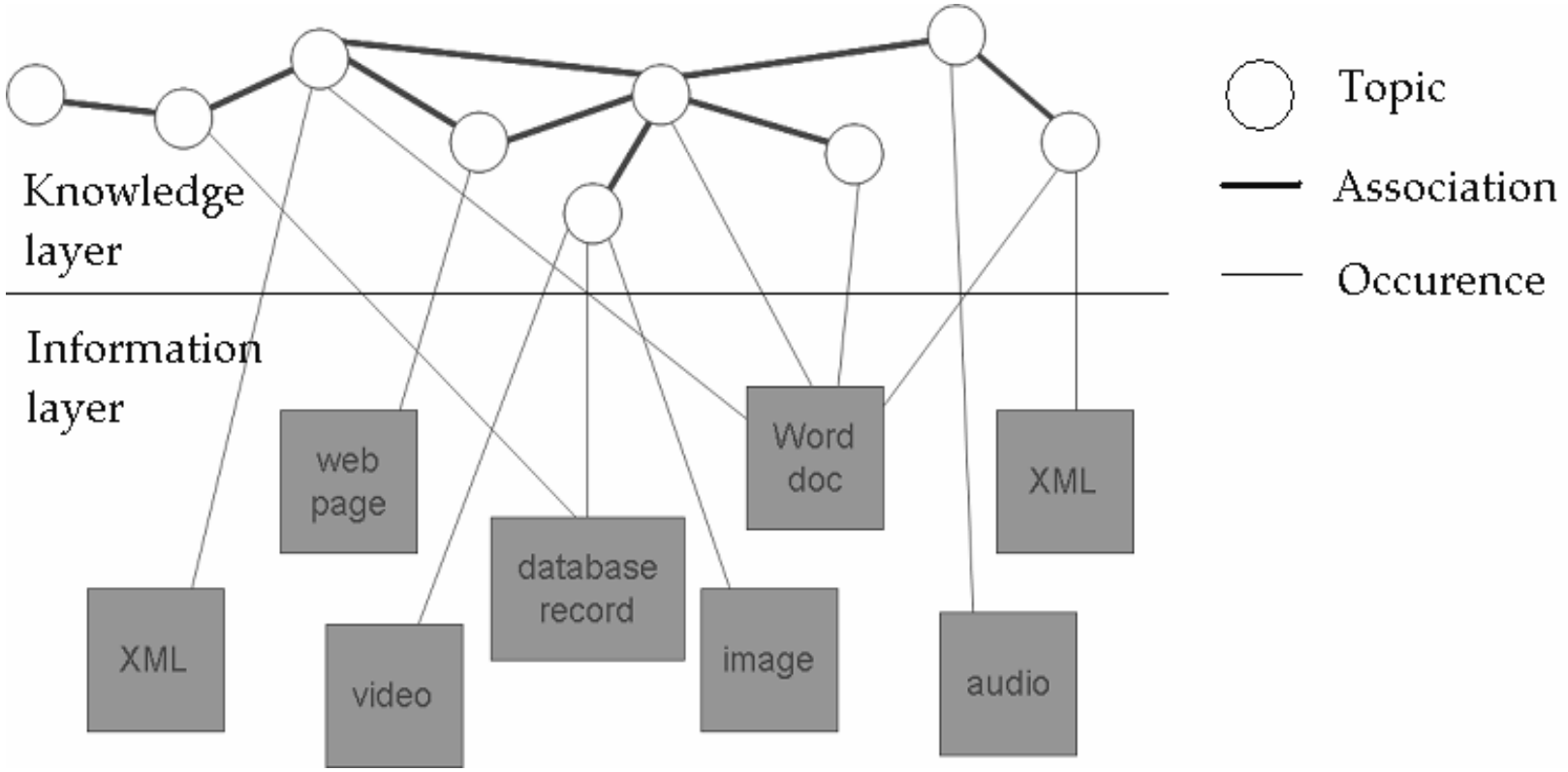
Taxonomy -> Thesauri -> Ontology

- **In a taxonomy**
  - BT/NT to build the hierarchy.
- **Thesauri**
  - BT/NT, RT, UF/USE relationships, and the SN property
- **Faceted classification does not really extend this language, but provides a consistent and useful discipline for applying it.**
- **With ontologies the language can be extended at will.**

Topic maps

- **subject-based classification technique**
- **The TAO of Topic Maps**
  - Topics, Associations and Occurrences
- **"topic" refers to the object or node in the topic map that represents the *subject* being referred to.**
- **Topics have types**
- **Topics have names, associations and occurrences**
- **Associations and occurrences can be typed**

Sample Topic Map

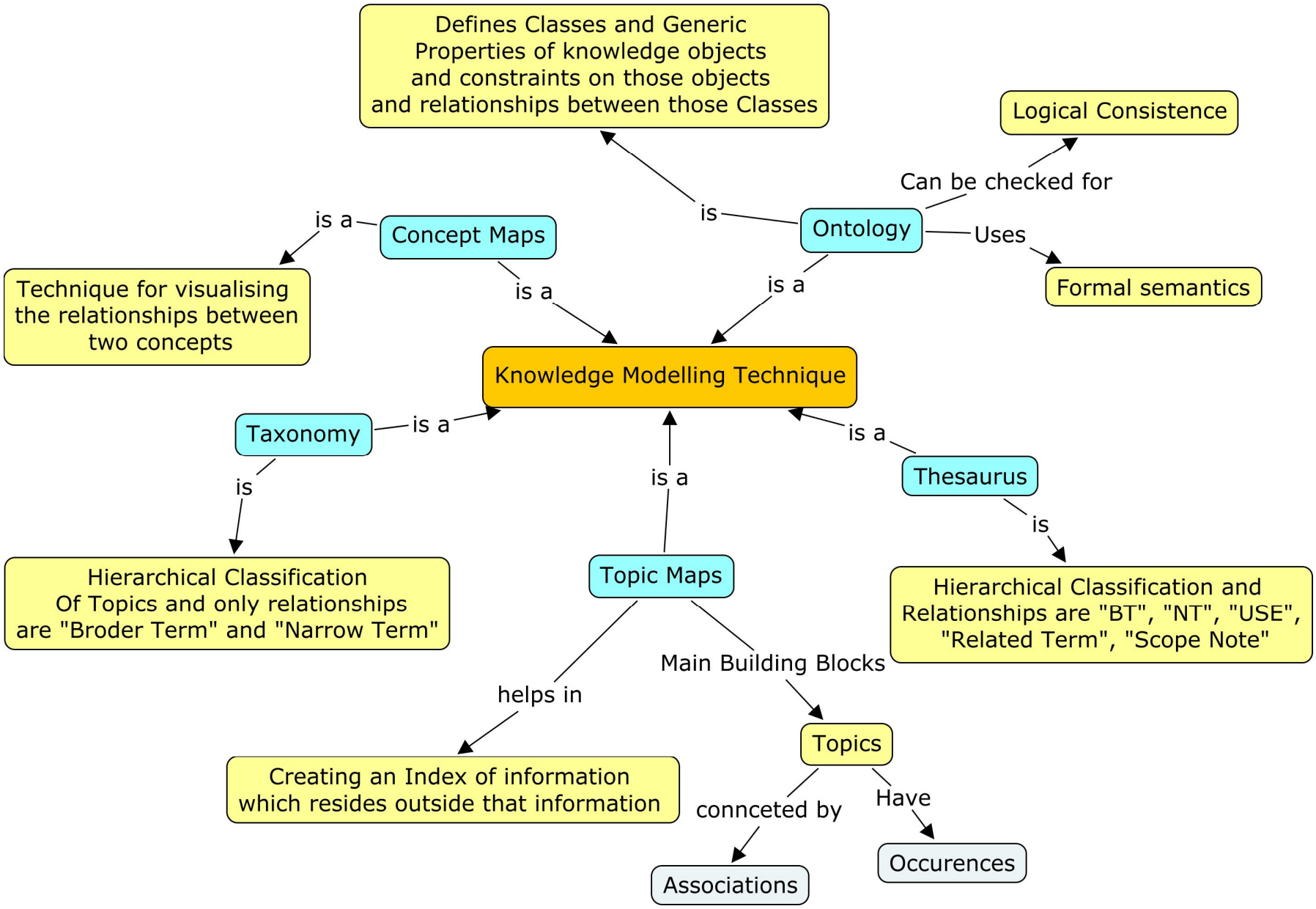


Ontologies and Topic Maps as KOS

- **A Knowledge Organisation System organises knowledge objects**
  - More or less everything can be considered a knowledge object

- **An Ontology**
  - Defines classes and generic properties of knowledge objects
  - Defines constraints on those classes and properties
  - Generally uses formal semantic declaration for those definitions
  - Can be checked for logical consistency

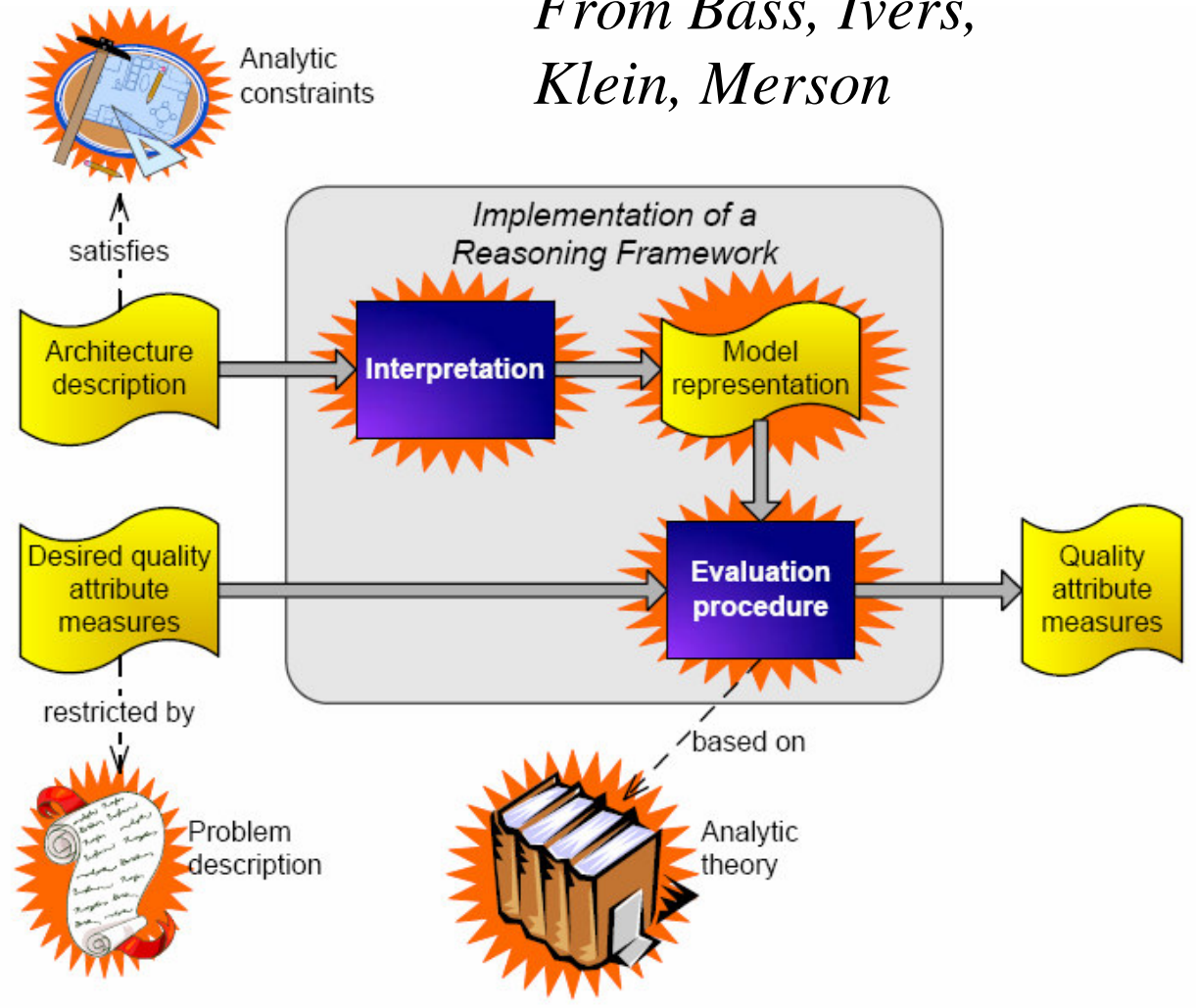
- **A Topic Map**
  - Defines a specific set of knowledge objects and their individual properties
  - Can include topics used as classes of the above
  - Has no formal semantics
  - Makes no provision for logical consistency


Defines Classes and Generic
Properties of knowledge objects
and constraints on those objects
and relationships between those Classes
Logical Consistence
Can be checked for
is
Ontology
Uses
Formal semantics
is a
Concept Maps
Technique for visualising
the relationships between
two concepts
is a
is a
Knowledge Modelling Technique
Taxonomy
is a
is
Hierarchical Classification
Of Topics and only relationships
are "Broder Term" and "Narrow Term"
is a
Topic Maps
is a
Thesaurus
is
Hierarchical Classification and
Relationships are "BT", "NT", "USE",
"Related Term", "Scope Note"
helps in
Main Building Blocks
Creating an Index of information
which resides outside that information
Topics
connceted by
Have
Associations
Occurences

Analytical models

- **Resoning frameworks**

From Bass, Ivers, Klein, Merson



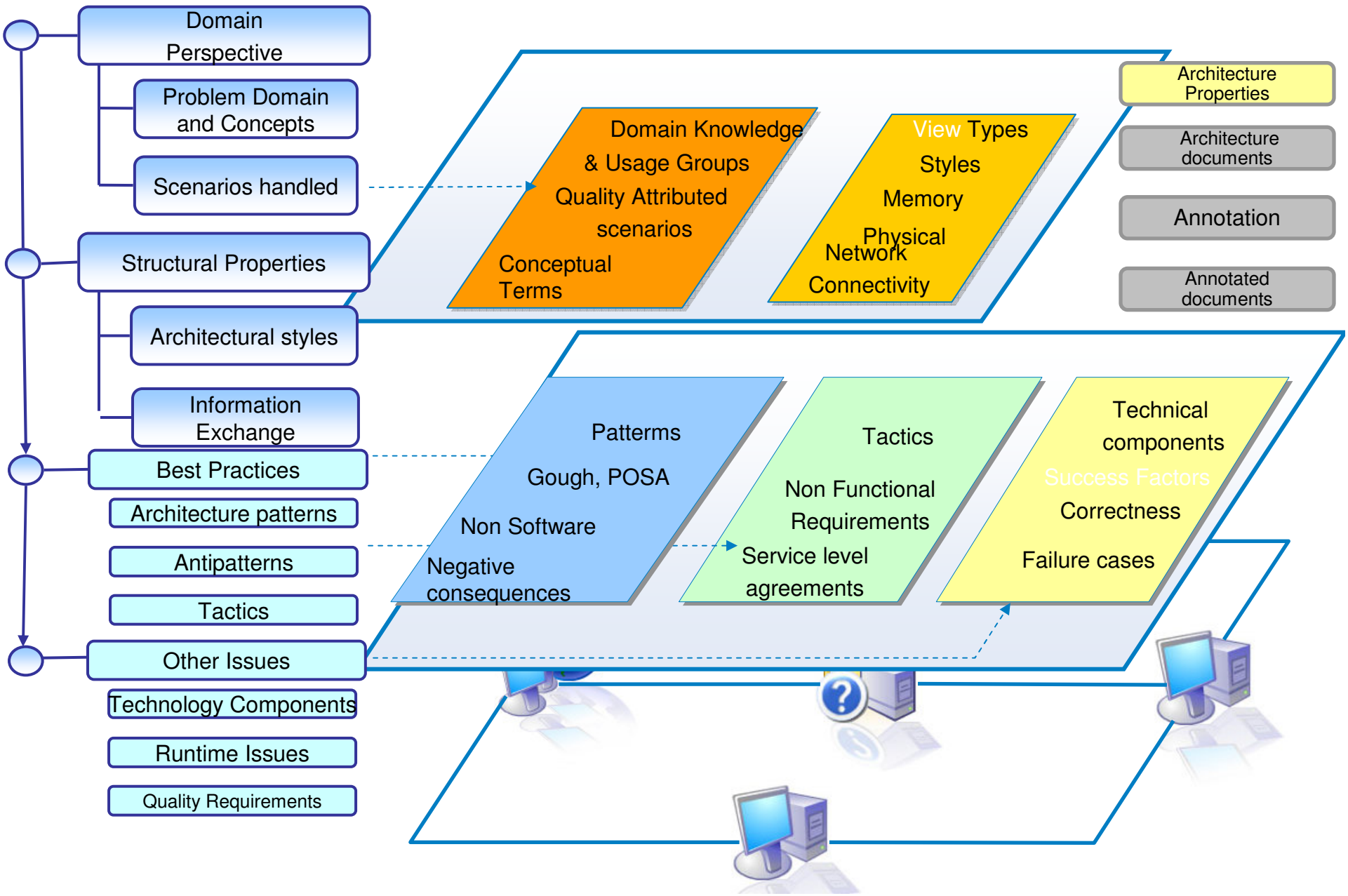
KB techniques in Architecture

- Document Retrieval
- Architecture Design
- Architecture Evaluation

Document Retrieval

- The problem of asset management and document retrieval
- Ontology driven search

Search Based on What - Architecture Properties



Architecture Properties

- Problem Domain
- Technology Components
- Structural Properties
- Architecture Tactics
- Information Exchange Mechanisms
- Architecture Patterns and Anti patterns
- Quality Requirements

Common Questions on Architecture Documents

- Search for arch documents which Apache web server with ODS gateway and CORBA middleware along with oracle backend?
- Retrieve the architecture documents with Interpreter and Rule based system styles?
- Did we use this tactic and what was the result?
- Retrieve the documents with solution architectures using MPI shared memory
- What is the consequence of using common interface and then wrapping to integrate?
- Retrieve the documents with similar problem domain ?

A Taxonomy or knowledge base for software architecture

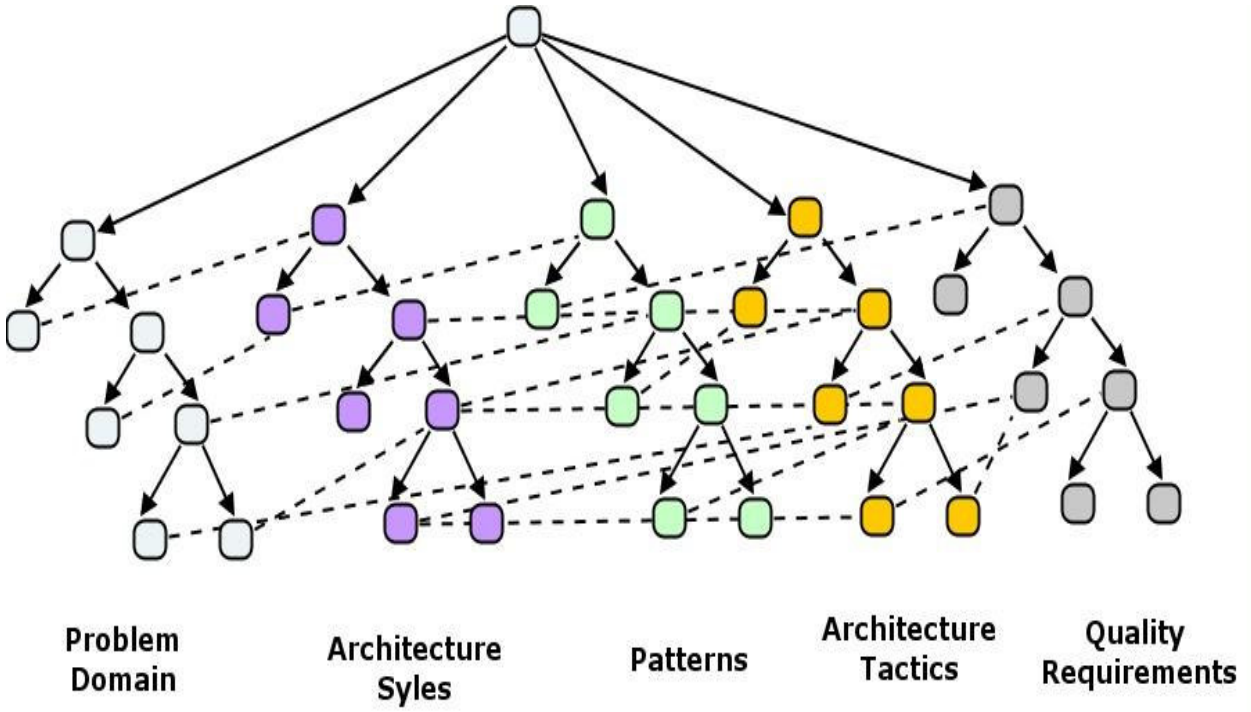
- Several relationships exist between architecture properties
- The relationships and the associated vocabulary forms a huge knowledge database
- Ontology of this knowledge base helps the architect in searching and retrieving architecture document in a semantic way .

Design of new systems using best practices

Pedagogy purposes

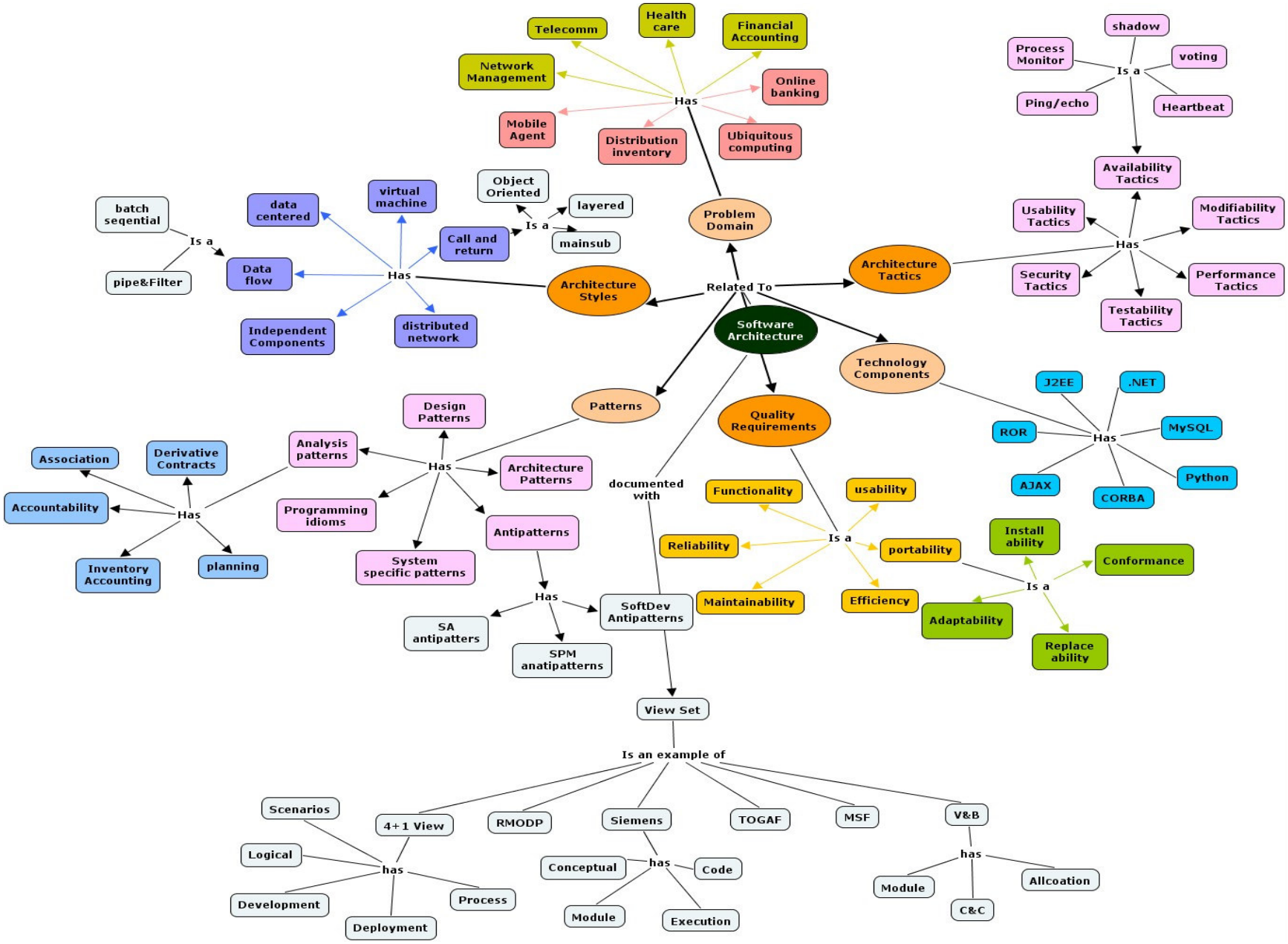
Searching a collection of architecture documents

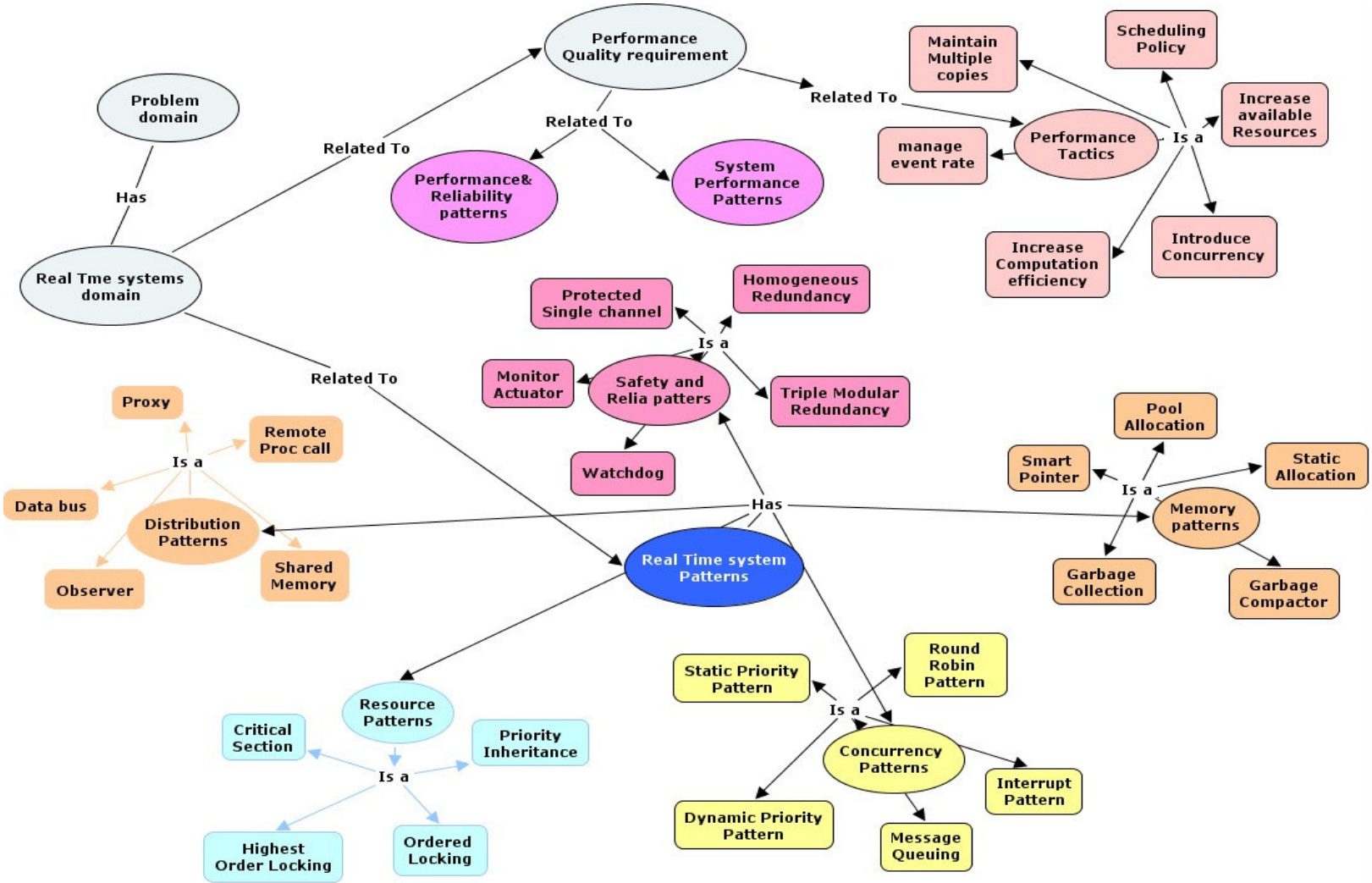
Conceptual overview of the ontology

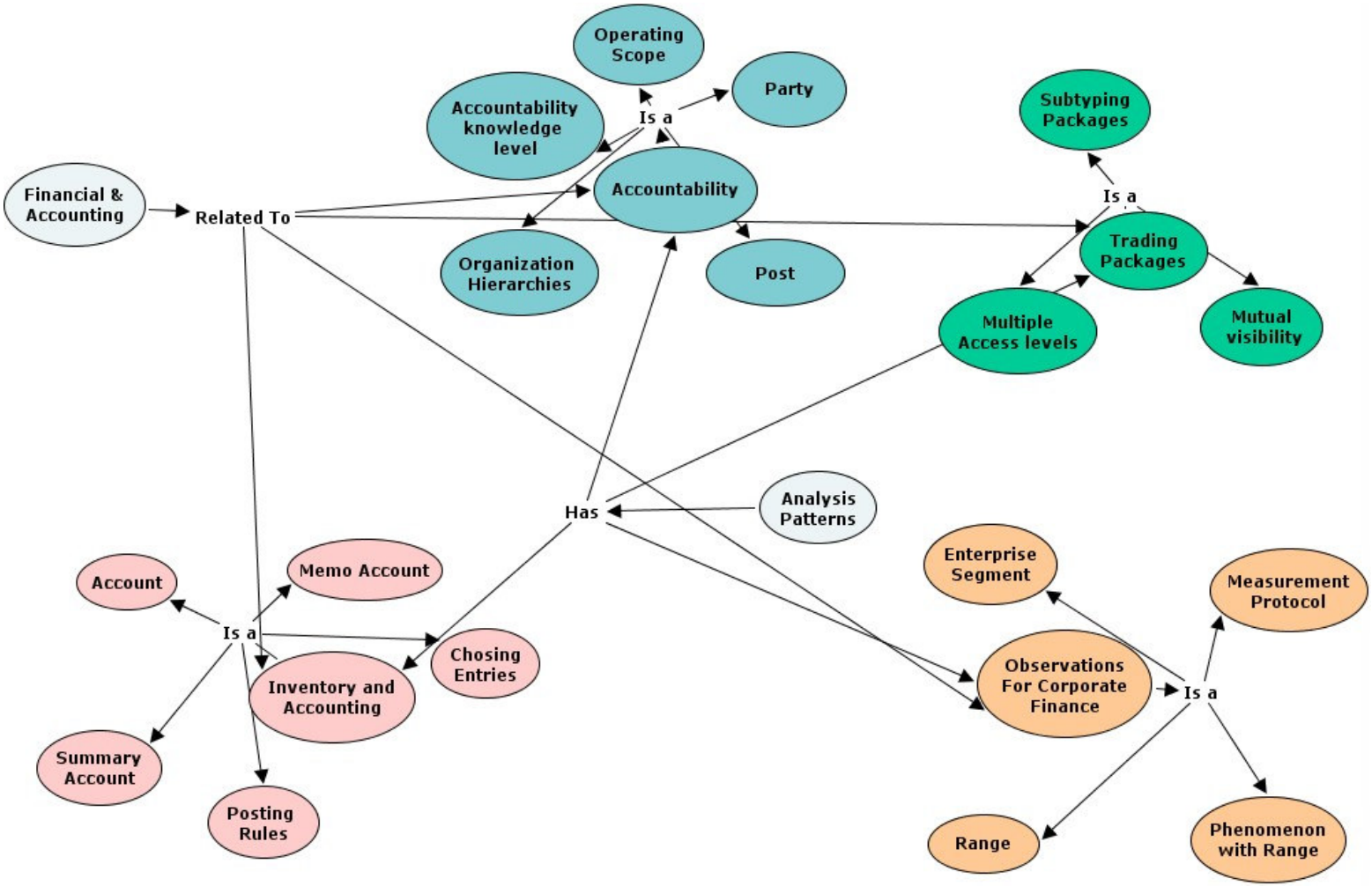


Relationships between Architecture Terms

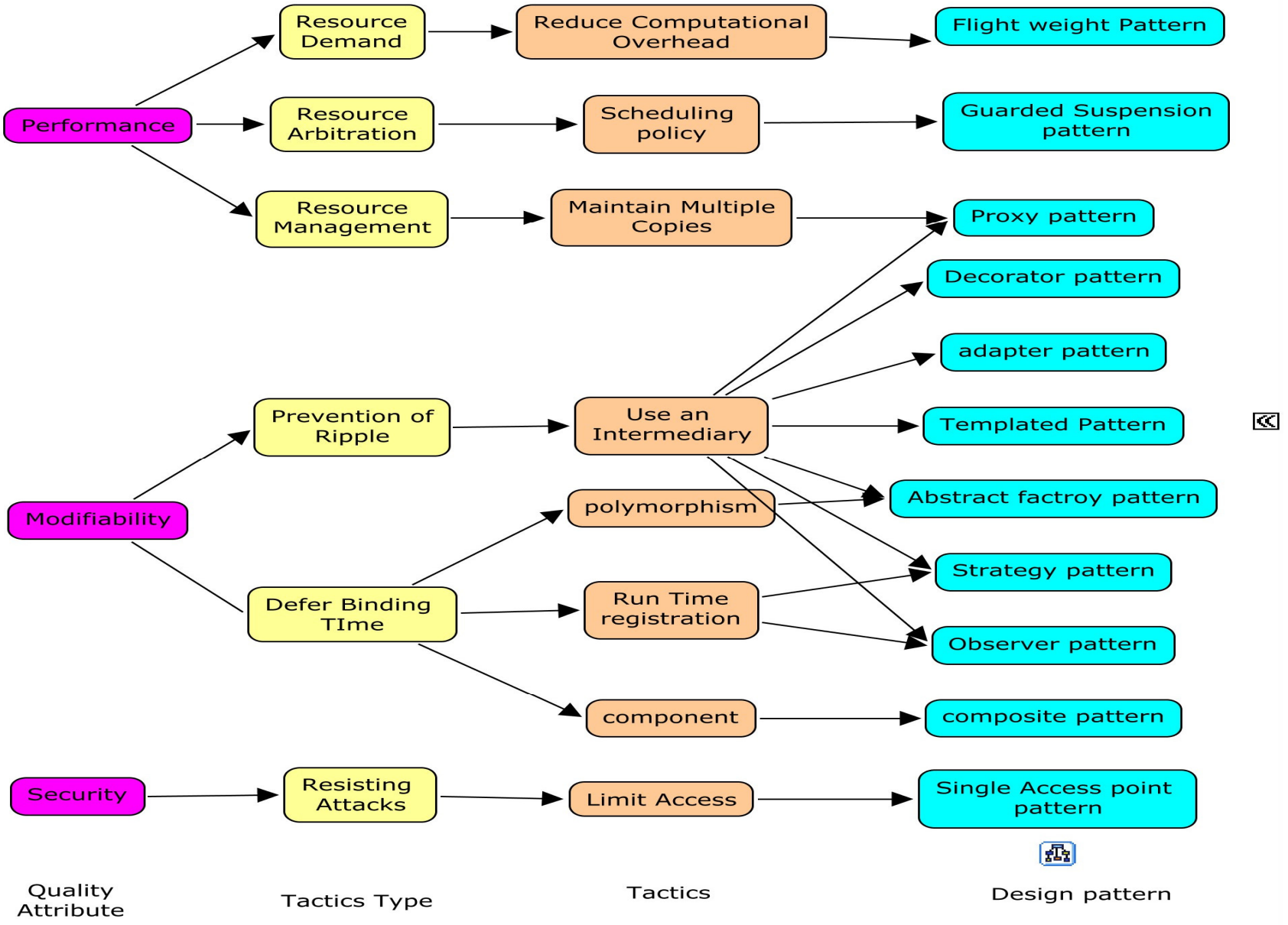
Name	Description
Used-to	**Denotes a means/ mechanism**
Related-To	**Denotes an association relationship**
Is-A	**Denotes a super, subclass relationship**
Is – part - of / Has	**Denotes an aggregation relationship**
Is-Similar-To	**Denotes an equivalence relationship**
Requires	**Denotes an association relationship**


Telecomm
Health care
Financial Accounting
Network Management
Online banking
Has
Mobile Agent
Distribution inventory
Ubiquitous computing
shadow
Process Monitor
voting
Is a
Ping/echo
Heartbeat
Availability Tactics
batch seqential
data centered
virtual machine
Object Oriented
layered
Is a
Call and return
mainsub
Problem Domain
Usability Tactics
Modifiability Tactics
Has
Security Tactics
Performance Tactics
Testability Tactics
pipe&Filter
Data flow
Has
Architecture Styles
Related To
Architecture Tactics
Independent Components
distributed network
Software Architecture
Technology Components
Patterns
Design Patterns
Quality Requirements
J2EE
.NET
ROR
MySQL
Has
Association
Derivative Contracts
Analysis patterns
Has
Architecture Patterns
AJAX
Python
CORBA
Accountability
Has
Programming idioms
Antipatterns
documented with
Functionality
usability
Reliability
Is a
portability
Install ability
Conformance
Inventory Accounting
planning
System specific patterns
Has
SoftDev Antipatterns
Maintainability
Efficiency
Is a
Adaptability
Replace ability
SA antipatters
SPM anatipatterns
View Set
Is an example of
Scenarios
4+1 View
RMODP
Siemens
TOGAF
MSF
V&B
Logical
has
Conceptual
has
Code
has
Module
Allcoation
Development
Process
Deployment
Module
Execution
C&C


Performance Quality requirement
Problem domain
Related To
Related To
Related To
Has
Performance& Reliability patterns
System Performance Patterns
Real Tme systems domain
Maintain Multiple copies
Scheduling Policy
Increase available Resources
manage event rate
Performance Tactics
Is a
Increase Computation efficiency
Introduce Concurrency
Protected Single channel
Homogeneous Redundancy
Is a
Monitor Actuator
Safety and Relia patters
Triple Modular Redundancy
Watchdog
Related To
Proxy
Remote Proc call
Is a
Data bus
Distribution Patterns
Observer
Shared Memory
Has
Real Time system Patterns
Pool Allocation
Smart Pointer
Is a
Static Allocation
Memory patterns
Garbage Collection
Garbage Compactor
Resource Patterns
Critical Section
Priority Inheritance
Is a
Highest Order Locking
Ordered Locking
Static Priority Pattern
Round Robin Pattern
Is a
Concurrency Patterns
Interrupt Pattern
Dynamic Priority Pattern
Message Queuing


Operating Scope
Party
Accountability knowledge level
Is a
Financial & Accounting
Related To
Accountability
Subtyping Packages
Is a
Trading Packages
Organization Hierarchies
Post
Multiple Access levels
Mutual visibility
Has
Analysis Patterns
Account
Memo Account
Is a
Chosing Entries
Inventory and Accounting
Summary Account
Posting Rules
Enterprise Segment
Measurement Protocol
Observations For Corporate Finance
Is a
Range
Phenomenon with Range

Quality attribute, tactics, patterns



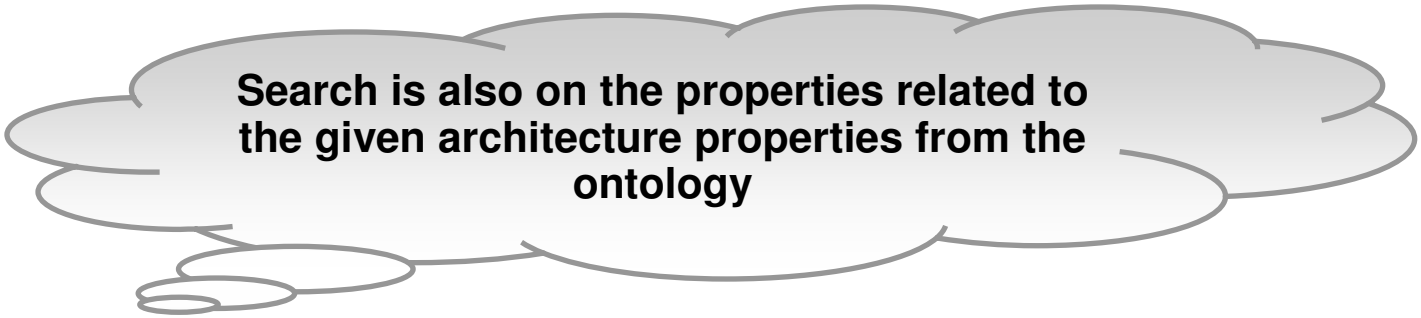
OntoSoftArch

- **OntoSoftArch is structured and controlled vocabulary for the software architecture to cover the basic set of architecture properties like styles, patterns, tactics, technologies etc..**
- **Some stats**
  - 1400 terms

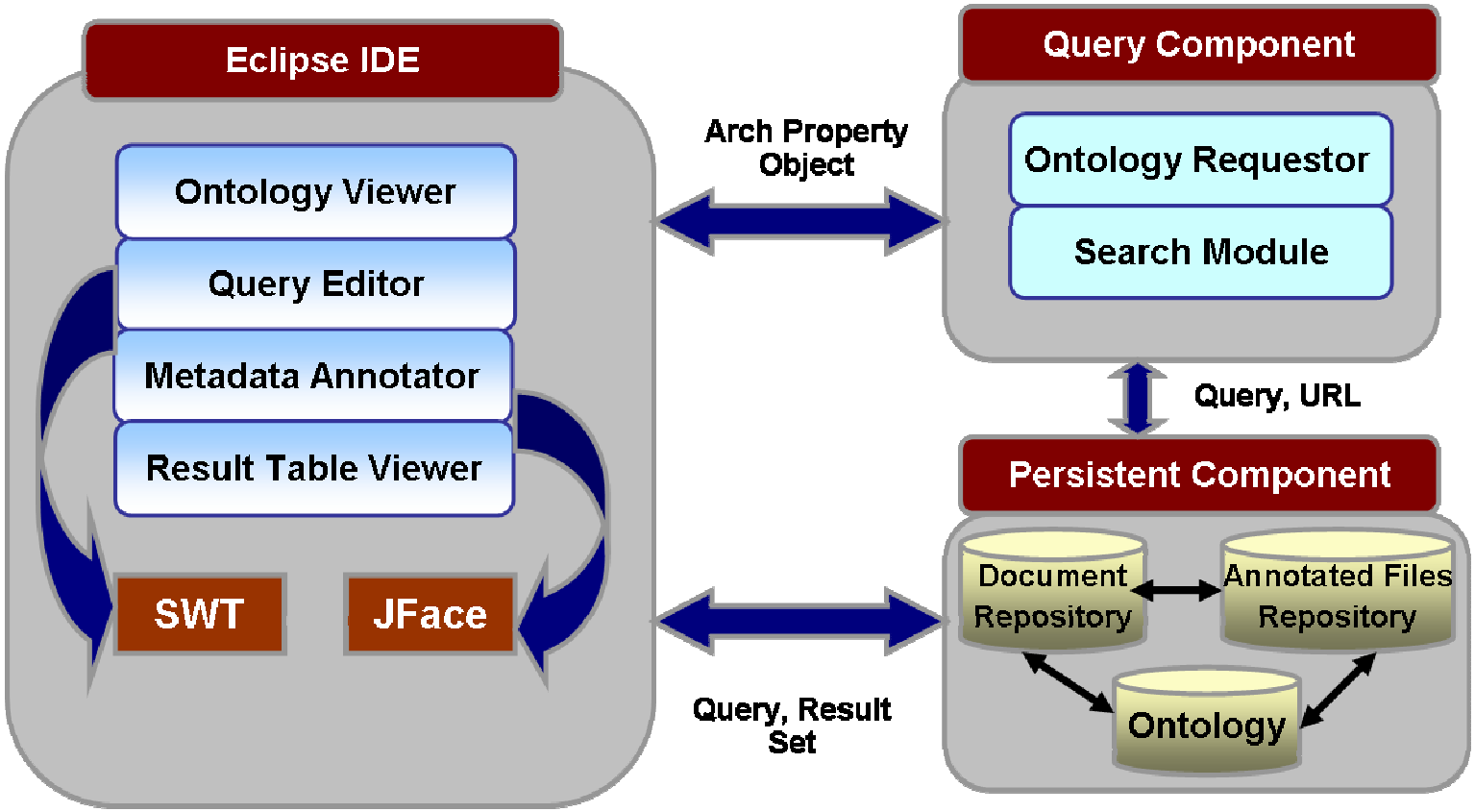
http://www.cse.iitk.ac.in/users/soft_arch/www/ontosoftarch/

Search For documents

- Parse the Architecture documents
- Search the Domain for the corresponding architecture properties
- Annotate the documents with architectural properties as XML file as shown below.
- Search in the XML files with architecture properties taken as input from the user.



System Architecture



Architecture Design Decisions

- Architecture knowledge is investigated in many projects –
- design decisions in the process of architecting
- Ex. Griffin Project, Kruchten..

Architectural Knowledge

- **Philippe Kruchten** – Taxonomy of architecture decisions and Building & exploiting architecture knowledge
- **GRIFFIN Project** – Knowledge modeling, representation, and acquisition approaches
- **Anton Jansen & Jan Bosch** – Describing architecture design decisions
- **Alessandro Garcia,Thaís Batista, Awais Rashid, Cláudio Sant'Anna** - Driving and Managing Architectural Decisions with Aspects
- **Ackerman and Tyree** - architecture is an instance of an ontology
- **Marco Sinnema, Salvador van der Ven,Sybren Deelstra** - Using Variability Modeling Principles to Capture Architectural Knowledge
- **Capilla, Nava, Perez,Duenas** – 4+1+1 View of architecture

Kruchten

- Taxonomy of design decisions
- Modeling - Not the design, But the decisions

Types of architectural design decisions

- **Ontocrises: existential decisions**
  - state what element will exist in the design
  - also anticrises (ban, things that will not exist)
- **Diacrises: property decisions**
  - associate predicate to set of elements
  - Cross cutting concerns, "aspects"
- **Pericrises: executive decisions**
  - constrain the above, the organization, etc.

Relationship between decisions

1. Constrains
2. Forbids
3. Enables
4. Subsumes
5. Conflicts with
6. Overrides
7. Comprises (is made of)
8. Is bound to
9. Is an alternative to
10. Is related too
11. Traces to
12. Does not comply with

Existence decisions ("ontocrises")

- **states that some element/artifact will be present in the system**
- ***structural decisions* and *behavioral decisions.***
- **Structural decisions:**
  - lead to the creation of subsystems, layers, partitions, components in some view of the architecture.
- **Behavioural decisions**
  - how the elements interact together to provide functionality or to satisfy some non functional requirement
- **Examples:**
  - The logical view is organized in 3 layers: Data layer, Business logic layer, User-Interface layer.
  - Communication between classes uses RMI (Remote Method Invocation).

Ban or non-existence ("anticrises")

- opposite of an existence decision
- states that some element will *not* appear in the design or implementation.
- The system does *not* use SQLServer as its relational database system.

Property decisions ("diacrises")

- **overarching trait or quality of the system.**
- **design rules or guidelines (when expressed positively) or design constraints (when expressed negatively),**
- **Properties are harder to trace to specific elements of the design or the implementation because they are often cross-cutting concerns, or they affect too many elements.**
- **Normally, they are implicit and forgotten.**
- ***Examples:***
  - All domain-related classes are defined in the Layer #2.
  - The implementation does not make use of open-source components whose license restricts closed redistribution.

Executive decisions ("pericrises")

- **do not relate directly to the design elements or their qualities,**
- **driven more by the business environment (financial), and affect the development process (methodological), the people (education and training), the organization**
- **Normally are about the choice of technologies and tools.**
- ***Examples:***
  - The database engine will be Oracle
  - The system will be developed using COBOL (the engineer won't leave me)

Sharing And Reusing architectural Knowledge (from a report by Patricia Lago)

- The **application-generic** knowledge,
  - what architects have implicitly in their heads, from their former experience in working in one or more domains.
  - a form of a "library" of knowledge - e.g. architectural patterns, tactics or reference architectures or even other software engineering techniques
  - can be generally applied in several applications independently of the domain.
- The **application-specific** knowledge,
  - of a specific application during the development or evolution of that application.
  - involves all the decisions that were taken during the architecting process of a particular system and the architectural solutions that implemented the decisions.

- Architecture Knowledge – product and process
- IEEE definition [IEEE 610.12-90], design is both "the process of defining the architecture, components, interfaces, and other characteristics of a system or component" and "the result of [that] process."

AK as a product

- concerns architectural knowledge *per se*
  - **Architectural design decisions,**
  - **architectural solutions (patterns, tactics, reference architectures)**
  - **and the mutual influence between decisions, solutions and the system quality attributes.**
- Meta-models, domain models, and conceptual models of architectural knowledge.
- Notations, languages and views for AK

AK as a **process**

- deals with *using* architectural knowledge during the software development lifecycle
- Use cases (or goals) of architectural knowledge.
- Tools, services, and application-generic infrastructure for supporting the use of architectural knowledge.
- Methods for discovery and reuse of architectural knowledge.
- Mechanisms combining architectural knowledge with other fields/disciplines (e.g. variability, aspect-oriented paradigm, versioning).

Modeling Architectural Knowledge

- **Representation is a function of the use**
- **Human consumption**
  - Reference, documentation, teaching, design guidance
- **Automated processing**
  - Search, expert system
- **Integration across 'architectures'**
  - Ontology
- **Application generic knowledge**
  - Tactics, patterns, scenarios, and the relationships between them
  - Ontology - search
  - Concept map - pedagogy
  - topic map – browse
  - Reasoning framework - archE
- **Application specific knowledge**
  - Design decisions (plan) – views, ontologies
  - What components, connectors… - views

Last Slide

- **What to teach**
  - Declarative – patterns, tactics, benchmarks
  - Episodic- make them do - all about training
- **Taxonomies, Ontology**
  - We need some standards
  - Will help in organization, visualization, search
  - Automatic annotation of legacy documents
  - "Codification and socialization of solutions, processes, etc"
- **New views – design rationale**
- **Ontology as a component**

References

- Shark Report – Patricia Lago
- M. Ali Babar & Ian Gorton – WICSA Tutorial
- Calero etal Ontologies for Software Engineering and Software Technology,, Springer 2006
- IEEE Software March 2006
- Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering, http://www.w3.org/2001/sw/BestPractices/SE/ODA/
- Javier Garzás, Mario Piattini, An Ontology for Microarchitectural Design Knowledge, IEEE Software march/April 2005

Thank you!