# Potential building blocks
## of a roadmap to S-BPM

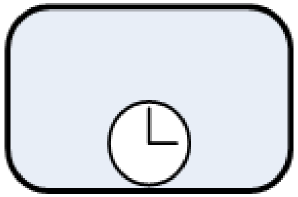
**Karlsruhe, October 22nd 2009**
**Hagen Buchwald, KIT, Institute AIFB**

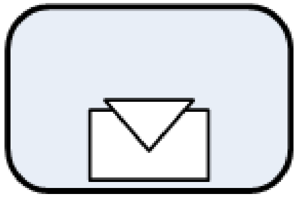
INSTITUTE OF APPLIED INFORMATICS AND
FORMAL DESCRIPTION METHODS (AIFB)

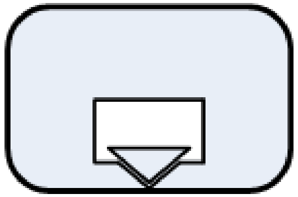
# Potential building blocks of a roadmap to S-BPM

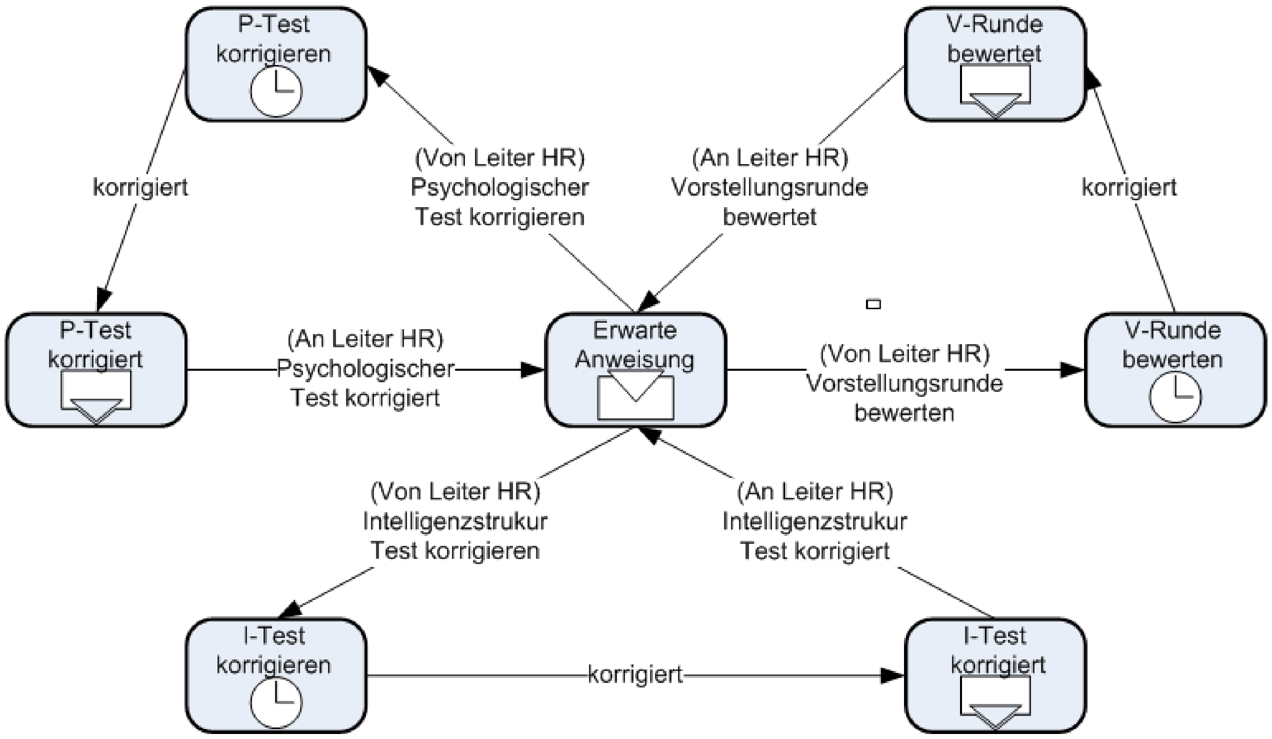
| Technology | Community | Methodology |
|---|---|---|
| T1<br>S-BPM Notation | C1<br>S-BPM Publication Series | M1<br>S-BPM Patterns |
| T2<br>S-BPM Architecture | C2<br>S-BPM ONE Conference | M2<br>S-BPM Process Life Cycle |
| T3<br>S-BPM Reference Implementation | C3<br>S-BPM Community Process | M3<br>S-BPM Maturity Levels |

# S-BPM Notation

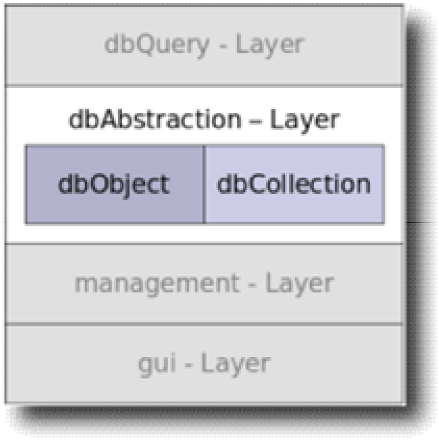







P-Test korrigieren
V-Runde bewertet
korrigiert
(Von Leiter HR) Psychologischer Test korrigieren
(An Leiter HR) Vorstellungsrunde bewertet
korrigiert
P-Test korrigiert
(An Leiter HR) Psychologischer Test korrigiert
Erwarte Anweisung
(Von Leiter HR) Vorstellungsrunde bewerten
V-Runde bewerten
(Von Leiter HR) Intelligenzstrukur Test korrigieren
(An Leiter HR) Intelligenzstrukur Test korrigiert
I-Test korrigieren
korrigiert
I-Test korrigiert

# S-BPM Architecture



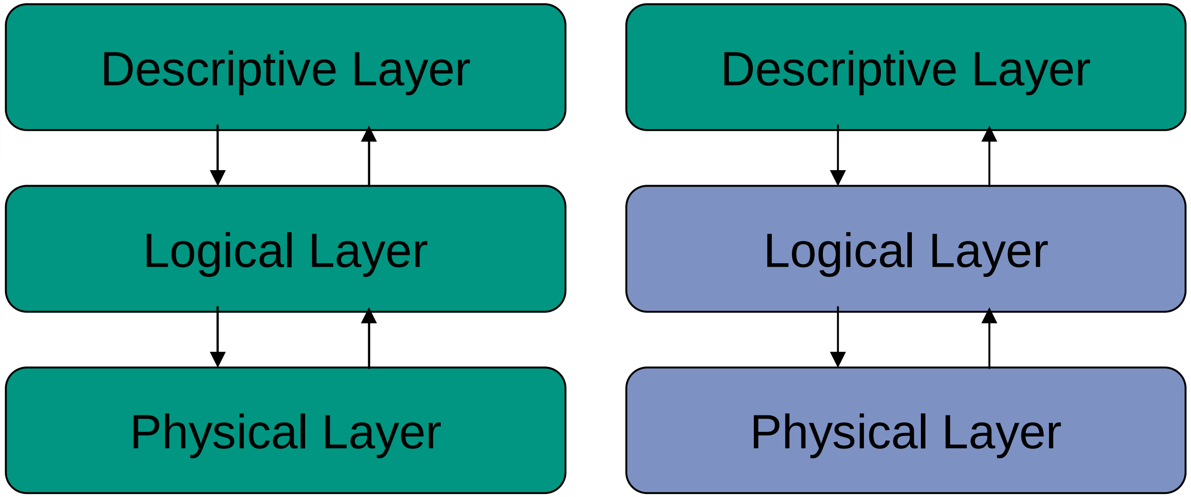
dbQuery - Layer

dbAbstraction – Layer

dbObject | dbCollection

management - Layer

gui - Layer



Descriptive Layer

Logical Layer

Physical Layer

Descriptive Layer

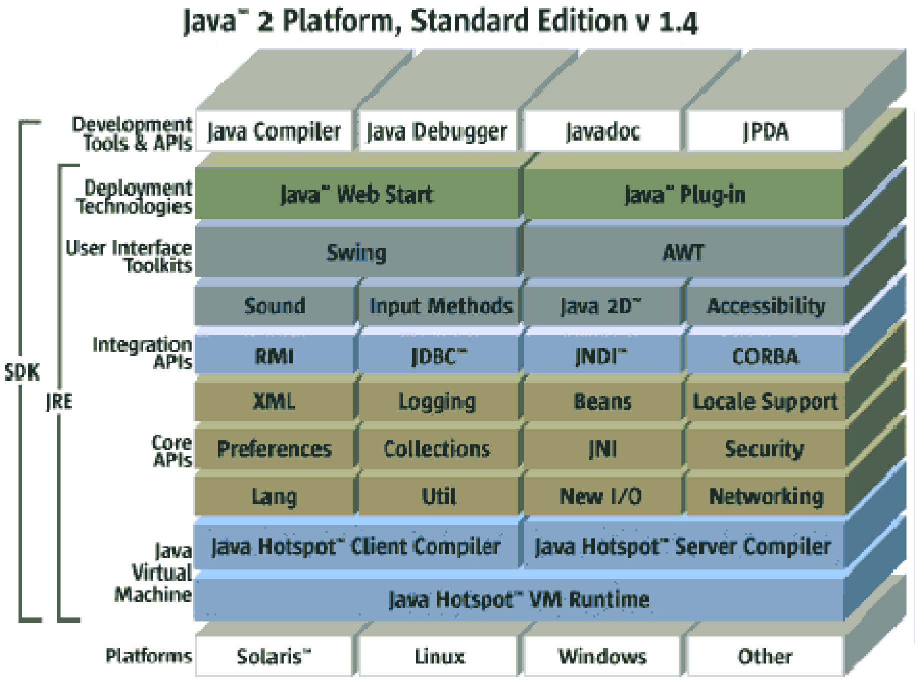
Logical Layer

Physical Layer

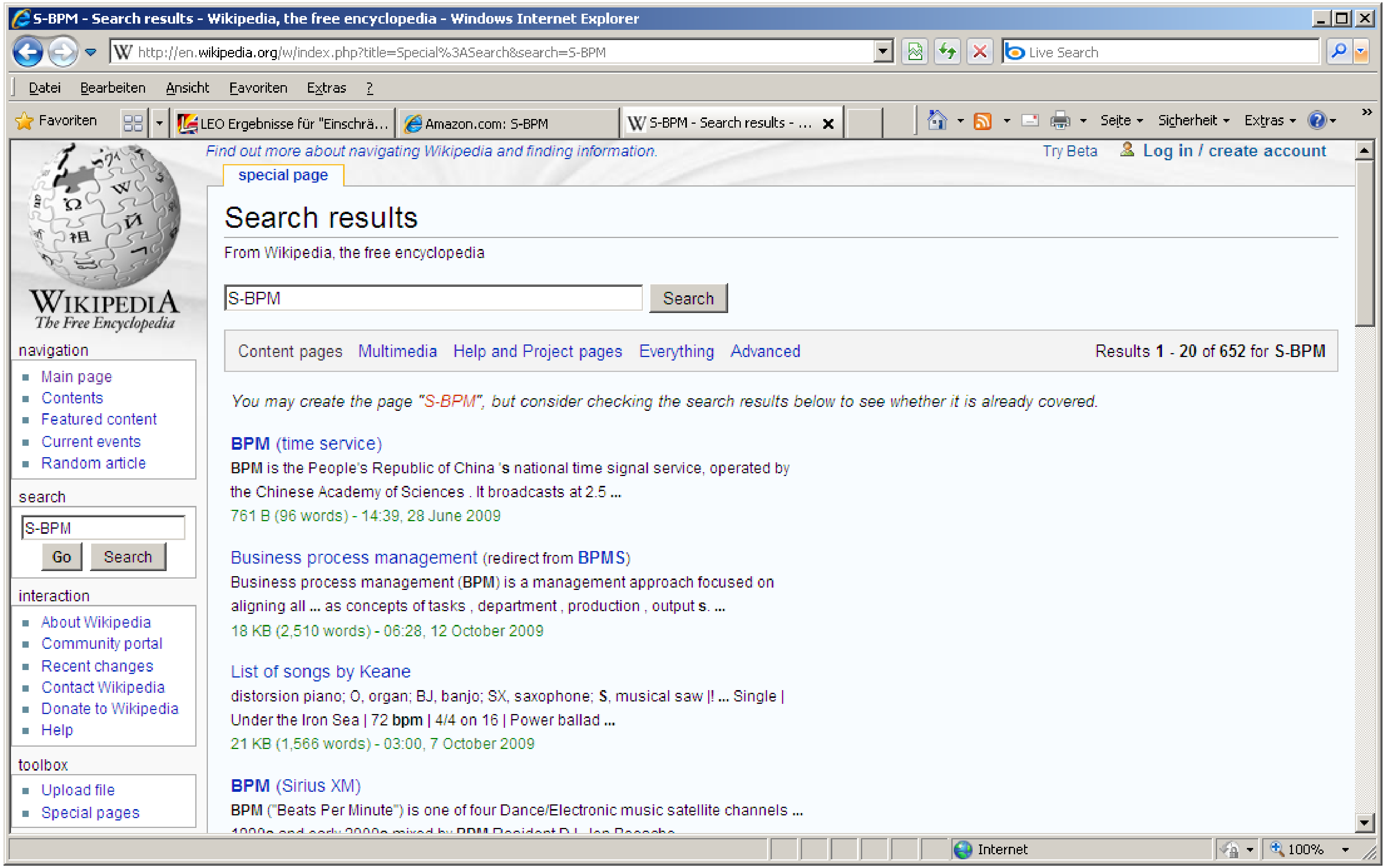
S-BPM platform A

S-BPM platform B

# S-BPM Reference Implementation



# S-BPM Publication Series (i.e. Wikipedia article)



Search results

From Wikipedia, the free encyclopedia

S-BPM Search

Content pages Multimedia Help and Project pages Everything Advanced Results **1** - **20** of **652** for **S-BPM**

*You may create the page "S-BPM", but consider checking the search results below to see whether it is already covered.*

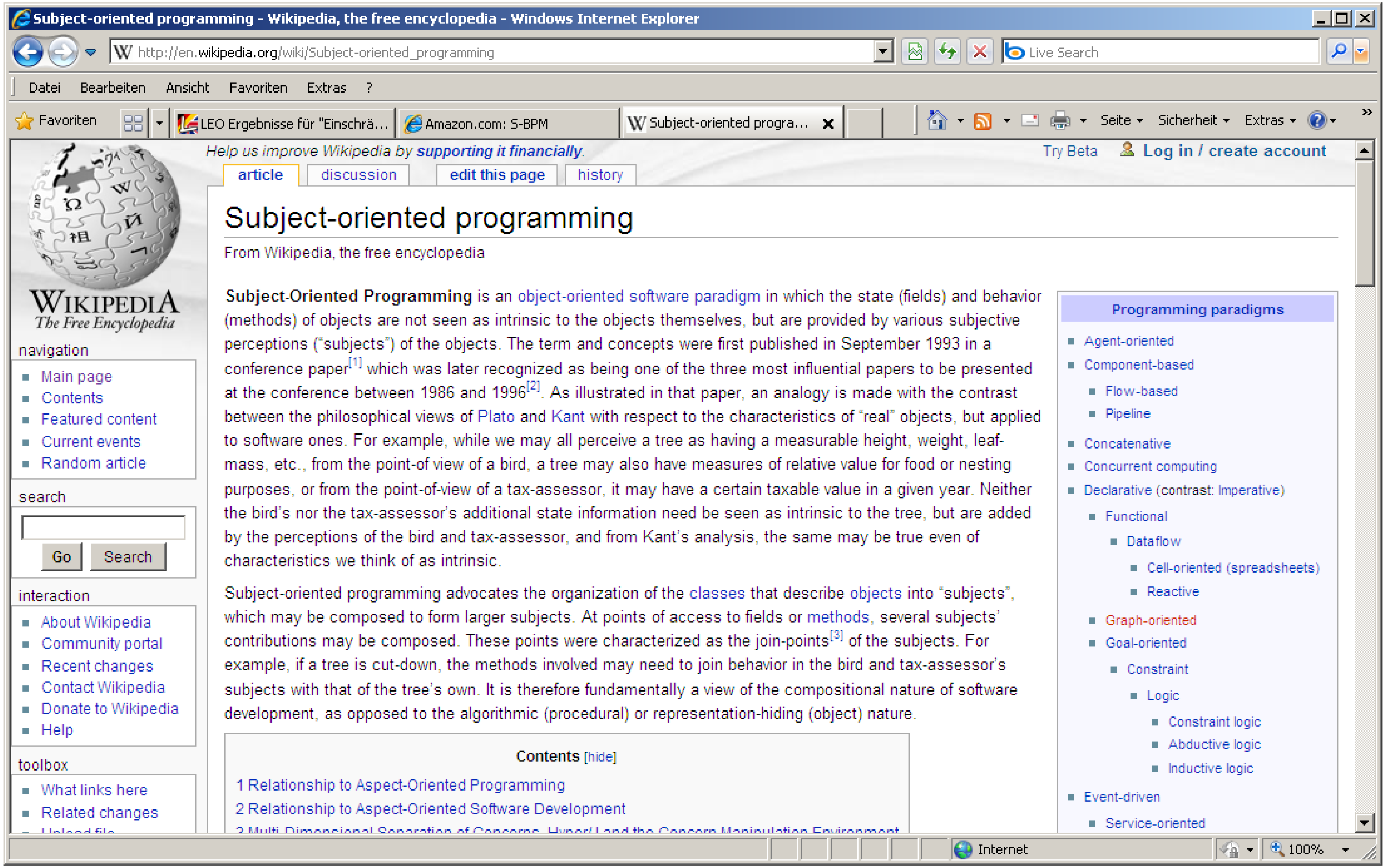
**BPM** (time service)
**BPM** is the People's Republic of China 's national time signal service, operated by the Chinese Academy of Sciences . It broadcasts at 2.5 ...
761 B (96 words) - 14:39, 28 June 2009

Business process management (redirect from **BPM S**)
Business process management (**BPM**) is a management approach focused on aligning all ... as concepts of tasks , department , production , output **s**. ...
18 KB (2,510 words) - 06:28, 12 October 2009

List of songs by Keane
distorsion piano; O, organ; BJ, banjo; SX, saxophone; **S**, musical saw |! ... Single | Under the Iron Sea | 72 **bpm** | 4/4 on 16 | Power ballad ...
21 KB (1,566 words) - 03:00, 7 October 2009

**BPM** (Sirius XM)
**BPM** ("Beats Per Minute") is one of four Dance/Electronic music satellite channels ...

# S-BPM Publication Series (i.e. Wikipedia article)



## Subject-oriented programming

From Wikipedia, the free encyclopedia

**Subject-Oriented Programming** is an object-oriented software paradigm in which the state (fields) and behavior (methods) of objects are not seen as intrinsic to the objects themselves, but are provided by various subjective perceptions ("subjects") of the objects. The term and concepts were first published in September 1993 in a conference paper[1] which was later recognized as being one of the three most influential papers to be presented at the conference between 1986 and 1996[2]. As illustrated in that paper, an analogy is made with the contrast between the philosophical views of Plato and Kant with respect to the characteristics of "real" objects, but applied to software ones. For example, while we may all perceive a tree as having a measurable height, weight, leaf-mass, etc., from the point-of view of a bird, a tree may also have measures of relative value for food or nesting purposes, or from the point-of-view of a tax-assessor, it may have a certain taxable value in a given year. Neither the bird's nor the tax-assessor's additional state information need be seen as intrinsic to the tree, but are added by the perceptions of the bird and tax-assessor, and from Kant's analysis, the same may be true even of characteristics we think of as intrinsic.

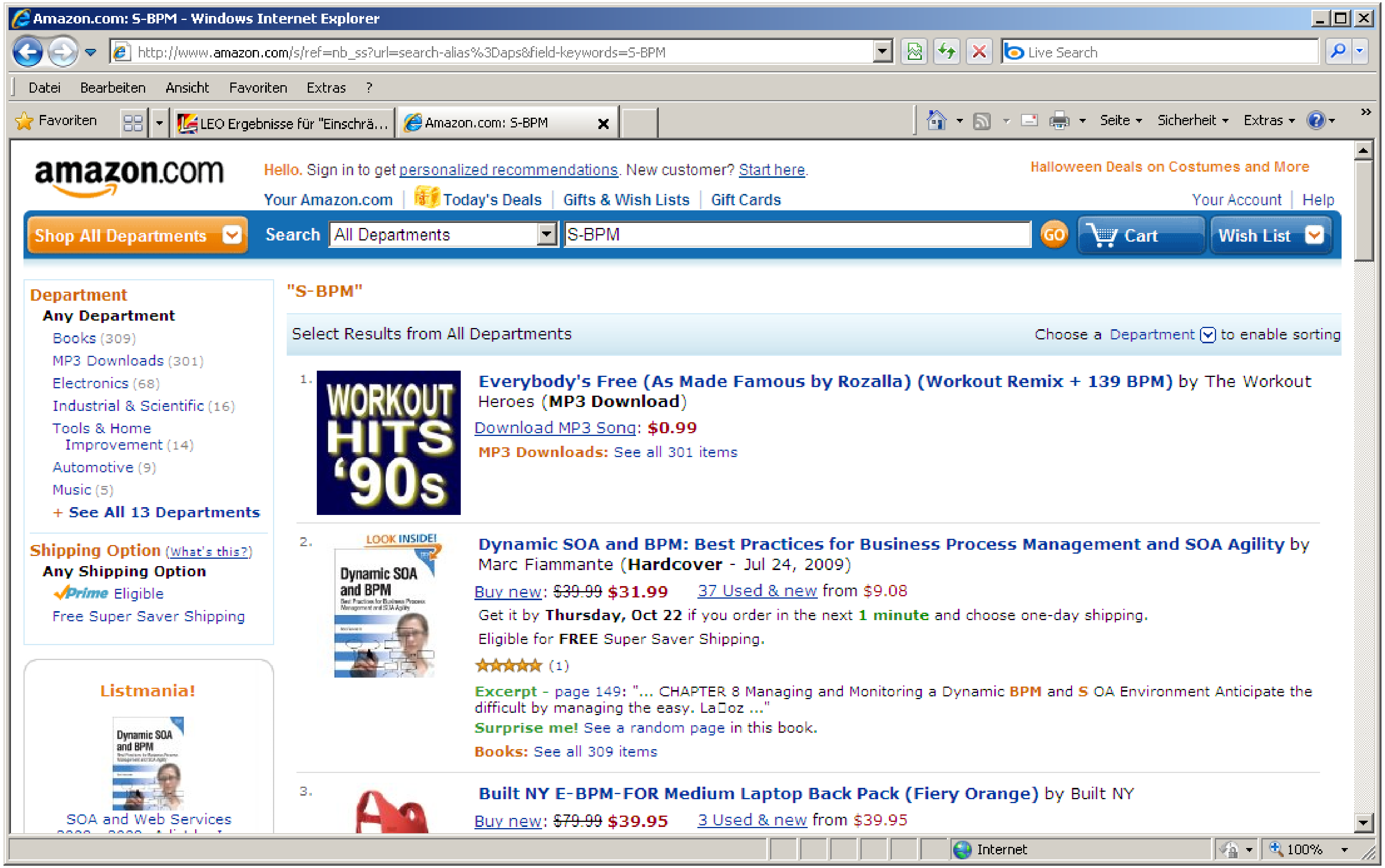
Subject-oriented programming advocates the organization of the classes that describe objects into "subjects", which may be composed to form larger subjects. At points of access to fields or methods, several subjects' contributions may be composed. These points were characterized as the join-points[3] of the subjects. For example, if a tree is cut-down, the methods involved may need to join behavior in the bird and tax-assessor's subjects with that of the tree's own. It is therefore fundamentally a view of the compositional nature of software development, as opposed to the algorithmic (procedural) or representation-hiding (object) nature.

Contents [hide]

**Programming paradigms**

- Agent-oriented
- Component-based
  - Flow-based
  - Pipeline
- Concatenative
- Concurrent computing
- Declarative (contrast: Imperative)
  - Functional
    - Dataflow
      - Cell-oriented (spreadsheets)
      - Reactive
  - Graph-oriented
  - Goal-oriented
    - Constraint
      - Logic
        - Constraint logic
        - Abductive logic
        - Inductive logic
- Event-driven
  - Service-oriented

# S-BPM Publication Series (i.e. Book series)



Hello. Sign in to get personalized recommendations. New customer? Start here.

Halloween Deals on Costumes and More

**Department**

**Any Department**

- Books (309)
- MP3 Downloads (301)
- Electronics (68)
- Industrial & Scientific (16)
- Tools & Home Improvement (14)
- Automotive (9)
- Music (5)
- + See All 13 Departments

**Shipping Option** (What's this?)

**Any Shipping Option**

- ✓Prime Eligible
- Free Super Saver Shipping

**Listmania!**

SOA and Web Services

**"S-BPM"**

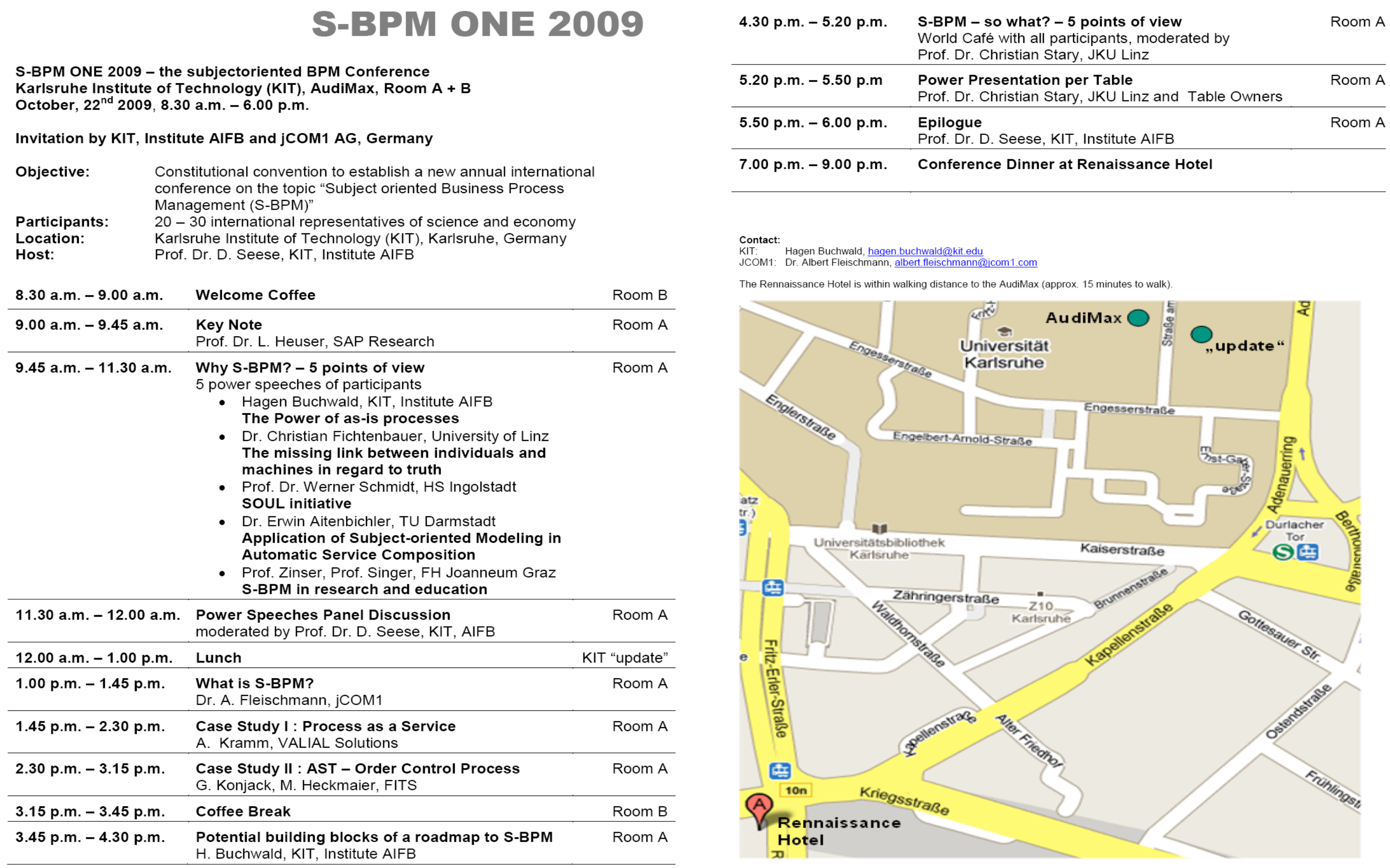
Select Results from All Departments — Choose a Department to enable sorting

1. **Everybody's Free (As Made Famous by Rozalla) (Workout Remix + 139 BPM)** by The Workout Heroes (**MP3 Download**)
Download MP3 Song: **$0.99**
**MP3 Downloads:** See all 301 items

2. **Dynamic SOA and BPM: Best Practices for Business Process Management and SOA Agility** by Marc Fiammante (**Hardcover** - Jul 24, 2009)
Buy new: ~~$39.99~~ **$31.99**   37 Used & new from $9.08
Get it by **Thursday, Oct 22** if you order in the next **1 minute** and choose one-day shipping.
Eligible for **FREE** Super Saver Shipping.
★★★★★ (1)
**Excerpt** - page 149: "... CHAPTER 8 Managing and Monitoring a Dynamic **BPM** and **S** OA Environment Anticipate the difficult by managing the easy. LaDoz ..."
**Surprise me!** See a random page in this book.
**Books:** See all 309 items

3. **Built NY E-BPM-FOR Medium Laptop Back Pack (Fiery Orange)** by Built NY
Buy new: ~~$79.99~~ **$39.95**   3 Used & new from $39.95

# S-BPM ONE



## S-BPM ONE 2009

**S-BPM ONE 2009 – the subjectoriented BPM Conference**
**Karlsruhe Institute of Technology (KIT), AudiMax, Room A + B**
**October, 22$^{nd}$ 2009, 8.30 a.m. – 6.00 p.m.**

**Invitation by KIT, Institute AIFB and jCOM1 AG, Germany**

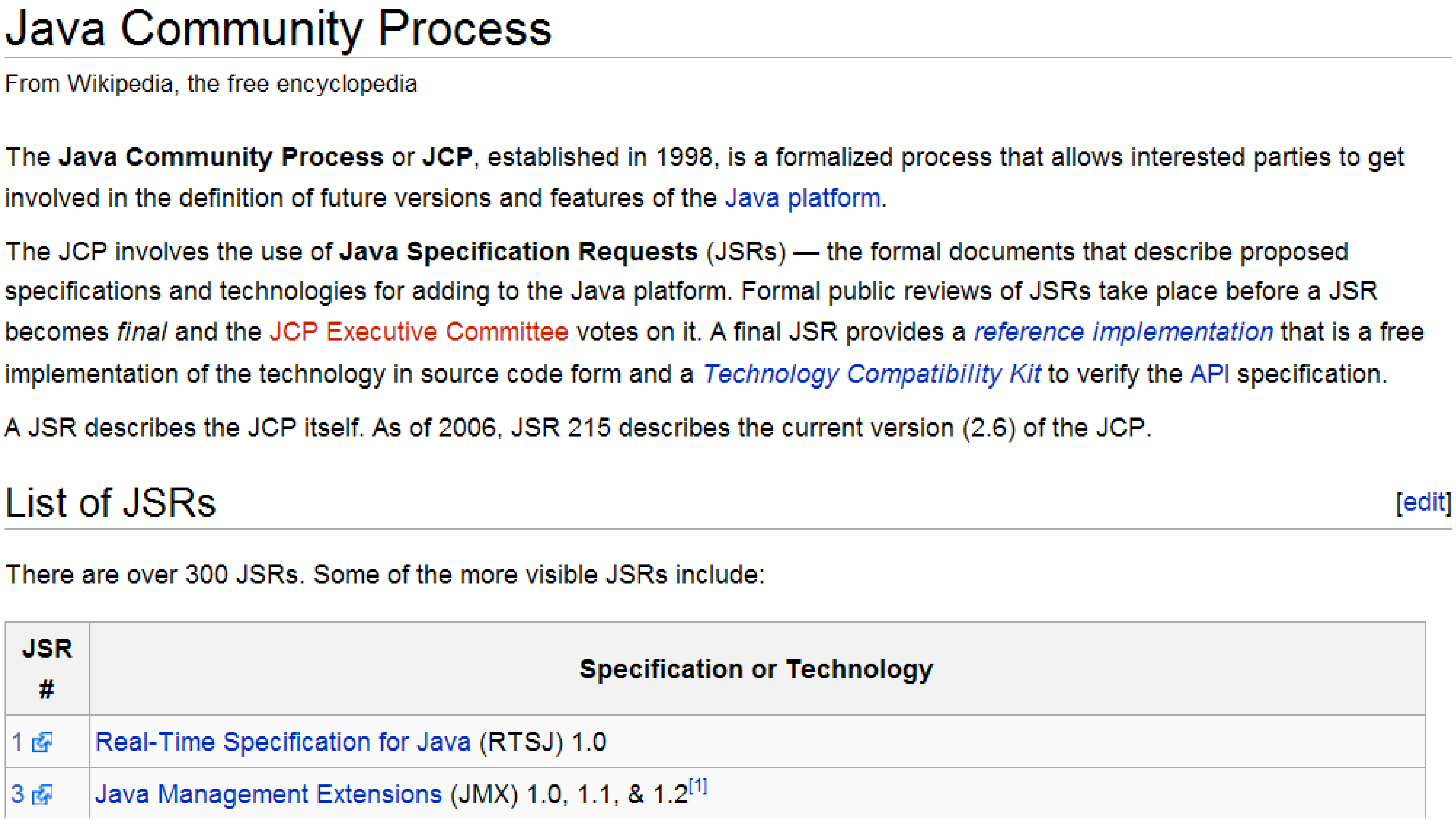
| | |
|---|---|
| **Objective:** | Constitutional convention to establish a new annual international conference on the topic “Subject oriented Business Process Management (S-BPM)” |
| **Participants:** | 20 – 30 international representatives of science and economy |
| **Location:** | Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany |
| **Host:** | Prof. Dr. D. Seese, KIT, Institute AIFB |

| | | |
|---|---|---|
| **8.30 a.m. – 9.00 a.m.** | **Welcome Coffee** | Room B |
| **9.00 a.m. – 9.45 a.m.** | **Key Note**<br>Prof. Dr. L. Heuser, SAP Research | Room A |
| **9.45 a.m. – 11.30 a.m.** | **Why S-BPM? – 5 points of view**<br>5 power speeches of participants<br>• Hagen Buchwald, KIT, Institute AIFB **The Power of as-is processes**<br>• Dr. Christian Fichtenbauer, University of Linz **The missing link between individuals and machines in regard to truth**<br>• Prof. Dr. Werner Schmidt, HS Ingolstadt **SOUL initiative**<br>• Dr. Erwin Aitenbichler, TU Darmstadt **Application of Subject-oriented Modeling in Automatic Service Composition**<br>• Prof. Zinser, Prof. Singer, FH Joanneum Graz **S-BPM in research and education** | Room A |
| **11.30 a.m. – 12.00 a.m.** | **Power Speeches Panel Discussion**<br>moderated by Prof. Dr. D. Seese, KIT, AIFB | Room A |
| **12.00 a.m. – 1.00 p.m.** | **Lunch** | KIT “update” |
| **1.00 p.m. – 1.45 p.m.** | **What is S-BPM?**<br>Dr. A. Fleischmann, jCOM1 | Room A |
| **1.45 p.m. – 2.30 p.m.** | **Case Study I : Process as a Service**<br>A. Kramm, VALIAL Solutions | Room A |
| **2.30 p.m. – 3.15 p.m.** | **Case Study II : AST – Order Control Process**<br>G. Konjack, M. Heckmaier, FITS | Room A |
| **3.15 p.m. – 3.45 p.m.** | **Coffee Break** | Room B |
| **3.45 p.m. – 4.30 p.m.** | **Potential building blocks of a roadmap to S-BPM**<br>H. Buchwald, KIT, Institute AIFB | Room A |
| **4.30 p.m. – 5.20 p.m.** | **S-BPM – so what? – 5 points of view**<br>World Café with all participants, moderated by Prof. Dr. Christian Stary, JKU Linz | Room A |
| **5.20 p.m. – 5.50 p.m** | **Power Presentation per Table**<br>Prof. Dr. Christian Stary, JKU Linz and  Table Owners | Room A |
| **5.50 p.m. – 6.00 p.m.** | **Epilogue**<br>Prof. Dr. D. Seese, KIT, Institute AIFB | Room A |
| **7.00 p.m. – 9.00 p.m.** | **Conference Dinner at Renaissance Hotel** | |

**Contact:**
KIT: Hagen Buchwald, hagen.buchwald@kit.edu
JCOM1: Dr. Albert Fleischmann, albert.fleischmann@jcom1.com

The Rennaissance Hotel is within walking distance to the AudiMax (approx. 15 minutes to walk).

# S-BPM Community Process



## Java Community Process

From Wikipedia, the free encyclopedia

The **Java Community Process** or **JCP**, established in 1998, is a formalized process that allows interested parties to get involved in the definition of future versions and features of the Java platform.

The JCP involves the use of **Java Specification Requests** (JSRs) — the formal documents that describe proposed specifications and technologies for adding to the Java platform. Formal public reviews of JSRs take place before a JSR becomes *final* and the JCP Executive Committee votes on it. A final JSR provides a *reference implementation* that is a free implementation of the technology in source code form and a *Technology Compatibility Kit* to verify the API specification.

A JSR describes the JCP itself. As of 2006, JSR 215 describes the current version (2.6) of the JCP.

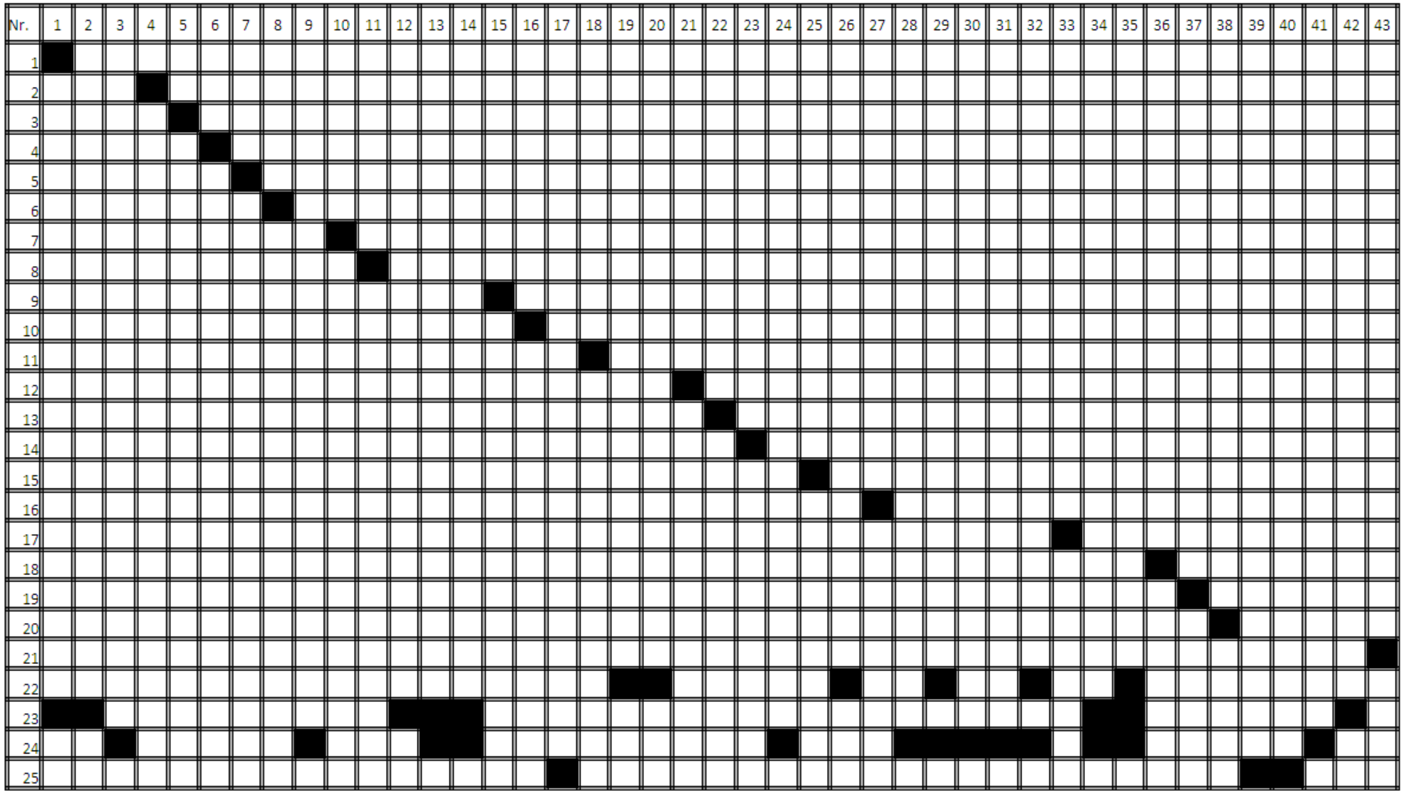
### List of JSRs

[edit]

There are over 300 JSRs. Some of the more visible JSRs include:

| JSR # | Specification or Technology |
|---|---|
| 1 | Real-Time Specification for Java (RTSJ) 1.0 |
| 3 | Java Management Extensions (JMX) 1.0, 1.1, & 1.2[1] |

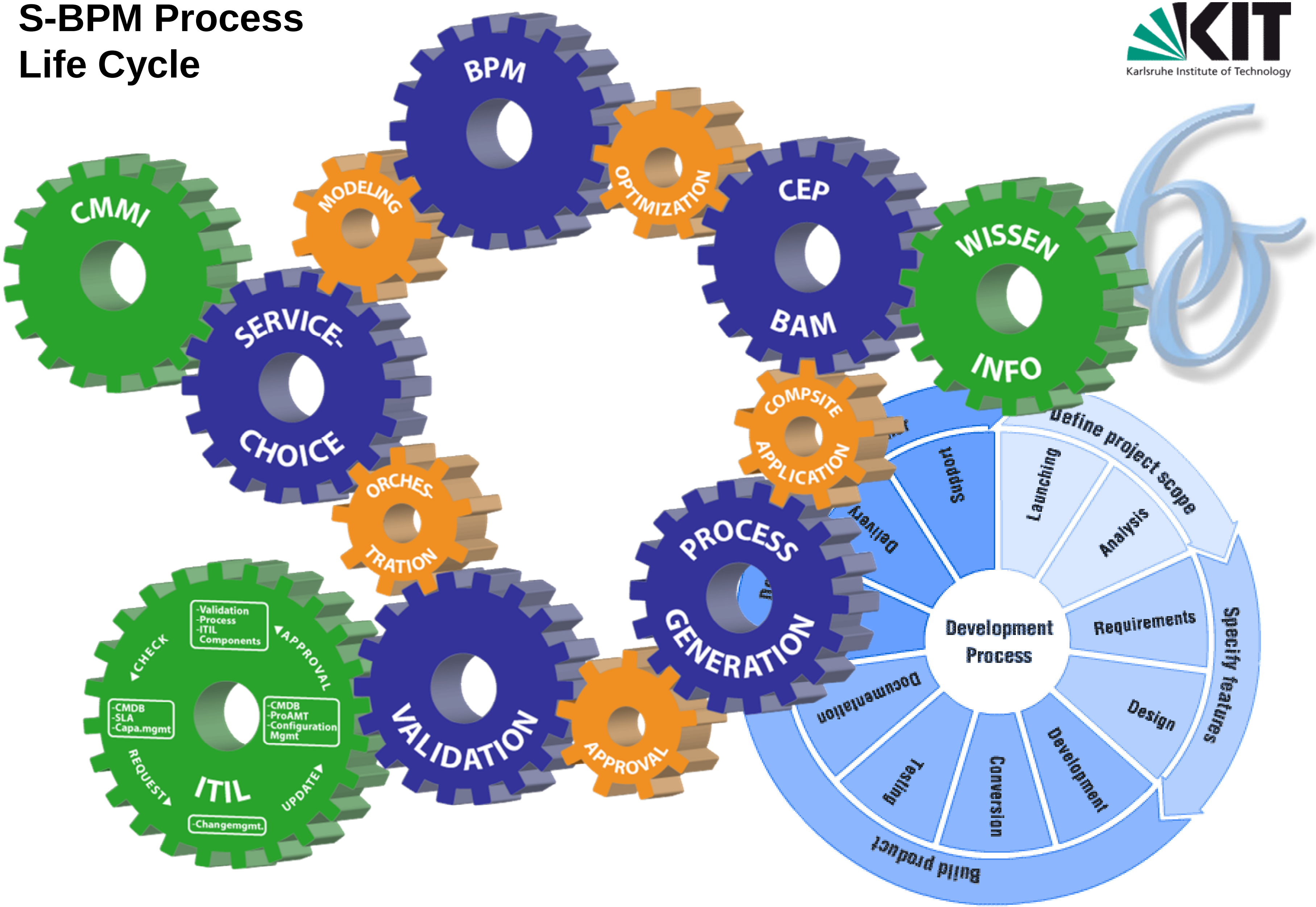
# S-BPM Patterns



| Nr. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ■ | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ■ | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ■ | | | | | | |
| 20 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ■ | | | | | |
| 21 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ■ |
| 22 | | | | | | | | | | | | | | | | | | | ■ | ■ | | | | | | ■ | | | ■ | | | ■ | | | ■ | | | | | | | | |
| 23 | ■ | ■ | | | | | | | | | | ■ | ■ | ■ | | | | | | | | | | | | | | | | | | | | ■ | ■ | | | | | | | ■ | |
| 24 | | | ■ | | | | | | ■ | | | | ■ | ■ | | | | | | | | | | ■ | | | | ■ | ■ | ■ | ■ | ■ | | ■ | ■ | | | | | | ■ | | |
| 25 | | | | | | | | | | | | | | | | | ■ | | | | | | | | | | | | | | | | | | | | | | ■ | ■ | | | |

- Nils Tölle and Norbert Graef showed in their Bachelor-Thesis 2009, that it is possible to reduce van der Aalst‘s 43 workflow patterns to 25 S-BPM patterns.

# S-BPM Process Life Cycle



# OOP Maturity Levels Bertrand Meyer, Object-Oriented Software Construction, 1986.

þ **Level 7**

Multiple inheritance

You may declare classes which inherit from more than one parent class.

þ **Level 6**

Polymorphism and dynamic binding

Elements of a system may reference objects of more than one class,
and routines may have different implementations in different classes.

þ **Level 5**

Inheritance

A class can be defined as an reduction or extension of another class.

þ **Level 4**

Classes

Classes are implementations of abstract data types (ADTs).

þ **Level 3**

Garbage collection

Unreferenced objects should be deallocated automatically
by the underlying runtime system.

þ **Level 2**

Data abstraction

Objects must be described as implementations of abstract data types (ADTs).

þ **Level 1**

Objectbased, modular structure

Systems a modularized based on their data structures.

# S-BPM Maturity Levels Bertrand Meyer, Object-Oriented Software Construction, 1986.

b **Level 7**

Multiple inheritance

You may declare classes which inherit from more than one parent class.

b **Level 6**

Polymorphism and dynamic binding

Elements of a system may reference subjects of more than one class, and behaviour may have different implementations in different classes.

b **Level 5**

Inheritance

A class can be defined as an reduction or extension of another subject.

b **Level 4**

Classes

Classes are implementations of abstract state machines (ASMs).

b **Level 3**

Linear scalability

The more cores the underlying hardware offers, the more subjects can act in parallel in order to respond to higher market demand.

b **Level 2**

Behaviour abstraction

Subjects must be described as implementations of abstract state machines (ASMs).

b **Level 1**

Subjectbased, modular structure

Systems a modularized based on their subject‘s behaviour (habits).