The *101 companies* community approach to knowledge engineering for software technologies and software languages

Ralf Lämmel  
*Software Languages Team*  
*University of Koblenz-Landau*
We have a problem!

Too much software technologies.
Too much software languages.
Too little time.

We need examples, abstractions, and analogies.
What’s 101companies?

The Hitchhiker's Guide to the Software Galaxy

Wannabe Wikipedia for Software Developers. :-)
What’s 101companies?

A community project aiming at a knowledge base about software technologies and languages based on implementations of a human-resources management system.
Related community resources
Wikipedia

A
- Action language
- Applicative programming language

C
- Compiled language

D
- Dependently typed programming
- Domain-specific language

E
- Esoteric programming language
- Expression-oriented programming language

F
- Fifth-generation programming language
- First-generation programming language
- Fourth-generation programming language
- Free-form language

G
- Glue language

H
- High-level programming language

I
- Intermediate language
- Interpreted language

L
- Line-oriented programming language
- List of educational programming languages
- Low-level programming language

P
- Pattern directed invocation programming language
- Persistent programming language
- Programming language generations
- Programming paradigm

Q
- Quantum programming

R

S
- Second-generation programming language

T
- Tactile programming language
- Third-generation programming language
- Toy language
- Transformation language

V
- Very high-level programming language
- Visual programming language
- Von Neumann programming languages

W
- User:Illusionoflife/typing
- Wide-spectrum language
- Write-only language
- Action language
- Applicative programming language

- Compiled language

- *Dependently typed programming*
- Domain-specific language

- Esoteric programming language
- Expression-oriented programming language

- Humus (programming language)
- Hybrid programming language

- Intermediate language
- Interpreted language

- Line-oriented programming language
- List of educational programming languages
- Low-level programming language

- Pattern directed invocation programming language
http://www.koders.com/
xmlRead.cpp

/* $Id: xmlRead.cpp,v 1.2 2004/03/14 09:16:14 zongo Exp $

** Ark - Libraries, Tools & Programs for MORPG developpements.

Language: C++
License: GPL
(C) 1999-2004 The Contributors of the Ark Project
LOC: 165
Gna! : Arkhart RPG Engine (project search) : .../arkhart/ArtTools/converters/ogrexml/xmlRead.cpp

VoiceXMLDocumentImpl.java

    } else {
        return new VoiceXMLElementImpl(this, null, tagName);
    }
}

Language: Java
(C) 1997-2000 DigitalSesame
LOC: 144
ObjectWeb Forge : Enhydra XMLC (project search) : .../wireless/voicxml/dom/xerces/VoiceXMLDocumentImpl.java

http://www.koders.com/
Google Code Search, Still Available

Google closed a lot of useful services to focus on Google+ and other core offerings. One of the best services that was recently shut down is Code Search, a search engine for open source code. If you go to google.com/codesearch, you'll see the following message:

"Sorry! Sadly, this service has been shut down. Much of Code Search's functionality is available at Google Code Hosting including search for Chromium. We're very sorry for the inconvenience."
99 Bottles of Beer

Language Python
(This example demonstrates the simplicity)

Date: 07/23/05
Author: Gerold Penz
URL: http://gerold.bcom.at/
Comments: 15
Info: http://www.python.org/
Score: ★★☆☆☆ (3.68 in 565 votes)

#!/usr/bin/env python
# -*- coding: iso-8859-1 -*-

99 Bottles of Beer (by Gerold Penz) Python can be simple, too :)

for quant in range(99, 0, -1):
    if quant > 1:
        print quant, "bottles of beer on the wall," , quant, "bottles of beer."
        if quant > 2:
            suffix = str(quant - 1) + " bottles of beer on the wall."
        else:
            suffix = "1 bottle of beer on the wall."
    elif quant == 1:
        print "1 bottle of beer on the wall, 1 bottle of beer."
        suffix = "no more beer on the wall!"
    print "Take one down, pass it around," , suffix
    print "--"
Benchmarks Game

(Redirected from Great language shootout)

This is for entries to The Computer Language Benchmarks Game (Formerly known as The Great Language Shootout). You can see the current Haskell results.

Haskell entries compared to C on the quad core, Feb 2009
http://rosettacode.org/
## Category: Solutions by Programming Task

Click on a programming task, and you can see solutions to that task in a variety of programming languages. If you want to see an unsorted list of all tasks on Rosetta Code, see [Programming Tasks](http://rosettacode.org/).

If you have an idea for a programming task not listed here, you may request it or add it yourself.

### Subcategories

This category has the following 58 subcategories, out of 58 total.

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategories</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3D</td>
</tr>
<tr>
<td>A</td>
<td>Animation</td>
</tr>
<tr>
<td></td>
<td>Arithmetic</td>
</tr>
<tr>
<td>B</td>
<td>Basic language learning</td>
</tr>
<tr>
<td>C</td>
<td>Checksums</td>
</tr>
<tr>
<td></td>
<td>Classic CS problems and programs</td>
</tr>
<tr>
<td></td>
<td>Compression</td>
</tr>
<tr>
<td></td>
<td>Concurrency</td>
</tr>
<tr>
<td></td>
<td>Conditional loops</td>
</tr>
<tr>
<td></td>
<td>Constructive Solid Geometry</td>
</tr>
<tr>
<td>G cont.</td>
<td>Geometric Primitives</td>
</tr>
<tr>
<td></td>
<td>Geometric Subtraction</td>
</tr>
<tr>
<td></td>
<td>Graphics algorithms</td>
</tr>
<tr>
<td></td>
<td>GUI</td>
</tr>
<tr>
<td>I</td>
<td>Image processing</td>
</tr>
<tr>
<td></td>
<td>Initialization</td>
</tr>
<tr>
<td></td>
<td>Internet Protocol</td>
</tr>
<tr>
<td></td>
<td>Iteration</td>
</tr>
<tr>
<td>J</td>
<td>Joystick</td>
</tr>
<tr>
<td>K</td>
<td>Keyboard Input</td>
</tr>
<tr>
<td>R</td>
<td>Randomness</td>
</tr>
<tr>
<td></td>
<td>Recursion</td>
</tr>
<tr>
<td></td>
<td>Regular expressions</td>
</tr>
<tr>
<td></td>
<td>Rosetta Code related</td>
</tr>
<tr>
<td>S</td>
<td>Sciences</td>
</tr>
<tr>
<td></td>
<td>Scope</td>
</tr>
<tr>
<td></td>
<td>Screen capture</td>
</tr>
<tr>
<td></td>
<td>Signal handling</td>
</tr>
<tr>
<td></td>
<td>Software Engineering</td>
</tr>
<tr>
<td></td>
<td>Sorting</td>
</tr>
<tr>
<td></td>
<td>Speech Recognition</td>
</tr>
<tr>
<td></td>
<td>Speech synthesis</td>
</tr>
<tr>
<td></td>
<td>Streams</td>
</tr>
<tr>
<td></td>
<td>String manipulation</td>
</tr>
</tbody>
</table>
G cont.

- Geometric Primitives
- Geometric Subtraction
- Graphics algorithms
- GUI

- Image processing
- Initialization
- Internet Protocol
- Iteration
Java Pet Store

Project Features
- Mailing Lists
- JIRA
- Subversion
- Wiki
- WikiHomePage

About this Project
Petstore *INACTIVE PROJECT* is a subproject of Blueprints *INACTIVE PROJECT*, was started in January 2011 and has 137 members. The project administrators are blueprints and shreedhar_ganapathy.

Join This Project

Last updated November 29, 2011 14:53, by sonyabarry

Please note: Sun stopped maintaining this project in 2007.
The project has been archived and is kept here for reference.
Sonya Barry Java.net Community Manager
EF 4.0 Hibernate like saveupdate

I have silverlight application, where I use Entity Framework(PostgreSQL) and WCF(not RIA). Here is the problem:

in database I have table organization and table of contacts. Organization has set of contacts.

EF entity is not a data contract used by wcf. I use convertor to make datacontract from entity and vice versa. So my question is how to save related entities like hibernate saveupdate(cascade="save-update")?
Please submit more entries:

DM to @101companies (Twitter)
101 companies:

What is it?

What’s it contribution?
What’s 101companies?

A community project aiming at a knowledge base about software technologies and languages based on implementations of a human-resources management system.

What code?
What knowledge?
What metadata?
Why is it called “101 companies”?

Company X: Swing + JDBC

Company Y: SWT + Hibernate

Company Z: GWT + MongoDB

...”101 ways of building a HRMS.”

or

“Building a HRMS for 101 companies.”
A Human Resources Management System

- Total salaries
- Increase salaries
- Cut salaries
- Edit employee data
- Import / export company data

© 2012, 101companies
Demo:

A 101 companies implementation using **HTML5** and local storage
Languages exercised

- Java
- C#
- VB.NET
- Python
- XML
- GIF
- Haskell
- XHTML
- .properties
- .ini
- HTML
- JSON
- Ecore
- PHP
- PHP4
- PHP5
- SQL
- .sh
- PNG
- ATL
- XSD
- Scala
- Text
- Markdown
- Prolog
- F#
- Perl
- CSS
- XMI
- AspectJ
- Smalltalk
- JAR
- Erlang
- WSDL
- Rascal
- Javascript
- 101meta
- Ruby
- C++
- Cobol
- (make)
- (Ant)
<table>
<thead>
<tr>
<th>Technologies exercised</th>
</tr>
</thead>
<tbody>
<tr>
<td>• make</td>
</tr>
<tr>
<td>• Ant</td>
</tr>
<tr>
<td>• Maven</td>
</tr>
<tr>
<td>• sbt</td>
</tr>
<tr>
<td>• JAXB</td>
</tr>
<tr>
<td>• xjc</td>
</tr>
<tr>
<td>• EF</td>
</tr>
<tr>
<td>• xsd.exe</td>
</tr>
<tr>
<td>• XmlSerializer</td>
</tr>
<tr>
<td>• ANTLR</td>
</tr>
<tr>
<td>• javac</td>
</tr>
<tr>
<td>• Eclipse</td>
</tr>
<tr>
<td>• Hibernate</td>
</tr>
<tr>
<td>• Java collections</td>
</tr>
<tr>
<td>• VS</td>
</tr>
<tr>
<td>• java.util</td>
</tr>
<tr>
<td>• java.io</td>
</tr>
<tr>
<td>• java.lang.reflect</td>
</tr>
<tr>
<td>• LINQ</td>
</tr>
<tr>
<td>• System.Xml</td>
</tr>
<tr>
<td>• System.Xml.Linq</td>
</tr>
<tr>
<td>• ADO</td>
</tr>
<tr>
<td>• SOA</td>
</tr>
<tr>
<td>• JPA</td>
</tr>
<tr>
<td>• EMFCompare</td>
</tr>
<tr>
<td>• 101explorer</td>
</tr>
<tr>
<td>• Mediawiki</td>
</tr>
<tr>
<td>• hackage</td>
</tr>
<tr>
<td>• cabal</td>
</tr>
<tr>
<td>• ghci</td>
</tr>
<tr>
<td>• ghc</td>
</tr>
<tr>
<td>• Swing</td>
</tr>
<tr>
<td>• AWT</td>
</tr>
<tr>
<td>• Java RMI</td>
</tr>
<tr>
<td>• JDBC</td>
</tr>
<tr>
<td>• zip</td>
</tr>
<tr>
<td>• jEdit</td>
</tr>
<tr>
<td>• jdom</td>
</tr>
<tr>
<td>• w3c.dom</td>
</tr>
<tr>
<td>• dom4j</td>
</tr>
<tr>
<td>• xom</td>
</tr>
<tr>
<td>• SAX</td>
</tr>
<tr>
<td>• ...</td>
</tr>
</tbody>
</table>

Mittwoch, 19. September 2012
Some technological spaces covered

- Objectware
- Modelware
- Grammarware
- XMLware
- Ontoware
- Tableware
- ...

Technological space
= Technology and community context in software engineering
The 101companies Repository

<table>
<thead>
<tr>
<th>#Files</th>
<th>Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>per implementation</td>
<td>LOC per implementation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Languages</th>
<th>442</th>
</tr>
</thead>
</table>

![Graph showing file and LOC distribution](image)

Mittwoch, 19. September 2012
Demo

101wiki -- the wiki of 101companies

http://101companies.org/
Consider 101companies’ value for knowledge acquisition

- 101companies contribution X ...
  - ... uses language L,
  - ... uses technology T,
  - ... implements feature F,
  - ... demonstrates concept C.

- Technology T helps with feature F.
- Technology $T_1$ always occurs together with technology $T_2$.
- Developer D has skills regarding language L and technology T.
- ...
Compare different efforts

<table>
<thead>
<tr>
<th>Aspect</th>
<th>101companies</th>
<th>Java Pet Store</th>
<th>99 Bottles of Beer</th>
<th>Rosetta Code</th>
<th>CLBG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Technologies</td>
<td>Java platform</td>
<td>Task</td>
<td>Tasks</td>
<td>Algorithms</td>
</tr>
<tr>
<td>Scope of comparison</td>
<td>Implementations</td>
<td>-</td>
<td>Languages</td>
<td>Languages</td>
<td>Languages</td>
</tr>
<tr>
<td>Technological spaces</td>
<td>covered</td>
<td>limited</td>
<td>ignored</td>
<td>limited</td>
<td>ignored</td>
</tr>
<tr>
<td>Ontology-driven</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Class room-tested</td>
<td>yes</td>
<td>yes(?)</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Source code</td>
<td>GitHub</td>
<td>Zip archive</td>
<td>Website</td>
<td>Website</td>
<td>Website</td>
</tr>
</tbody>
</table>

[http://softlang.uni-koblenz.de/101companies/inauguration/] (TOOLS 2012)
Software technologies
and software languages
in education
About 15,700,000 results (0.35 seconds)

Free Online Course Materials | Courses | MIT OpenCourseWare
ocw.mit.edu/courses/
Free MIT courses, including videos, audio, simulations, lecture notes, and exams. ... and Computer Science; Engineering Systems Division; Experimental Study ... Audio/Video Courses - Electrical Engineering and ... - Mathematics - Physics

Computer Science Online Courses, Computer ... - Free Video Lecture
freevideolectures.com/Subject/Computer-Science - United States
Free Computer Science video lectures, free Online Computer Science courses, Video Tutorials, University Courses, Youtube, Lecture notes, School Video ...

Lec 1 | MIT 6.00 Introduction to Computer Science and Programming ...
www.youtube.com/watch?v=k6U...
19 Aug 2009 - 54 min - Uploaded by MIT
Lecture 1: Goals of the course; what is computation; introduction to data types, operators ... Lec 1 | MIT 6.00 ...

More videos for lectures online computer science »
Programing Language Video Lectures - Free Science and Video ...
freescienceonline.blogspot.com/.../programming-language-vid...
30 Sep 2007 – This month I present to you my findings on programming language video lectures. I found some good introductory lectures on C++, Java, Perl, ...

Java (programming language) - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Java_(programming_language)
Java is a programming language originally developed by James Gosling at Sun .... of language features supporting better code analysis (such as inner classes, the ..... "Java - definition of Java by the Free Online Dictionary, Thesaurus and ...
You've visited this page many times. Last visit: 2/2/12

Programming Languages - Google Code University - Google Code
code.google.com/edu/languages/
This page contains classes, lectures, and tutorials about various computer programming languages. The technology lectures and classes provide a great way to ...
Subject areas in courses

• Bells and whistles of ... (e.g., Haskell)
• Programming language paradigms
• Good old compiler construction
• Domain-specific language engineering
• Foundations of ... (e.g., database systems)
• ...
Courses

- **Programming Techniques and Technologies**. Lecture at University Koblenz-Landau, Computer Science Department. Since 2010.
- **Modern data technologies**. Seminar (tutorial) first presented for Debeka employees in September 2012.
- **Modern web programming**. Seminar (tutorial) first presented for Debeka employees in September 2012.
Course Programming Techniques and Technologies @ Koblenz

- 05/03/2011: Grammar Programming (slides: .pdf, video: .mov)
- 05/05/2011: Generative Programming (slides: .pdf, video: .mov)
- 05/19/2011: Metaprogramming (slides: .pdf, video: .mov)
- 06/07/2011: Pattern-oriented Programming cont'd
- 06/21/2011: Multithreaded programming (slides: .pdf, video: .mov)
- 06/30/2011: Q&A (slides: .pdf, video: none)
- 07/05/2011: Hackathon -- Part I
- 07/07/2011: Hackathon -- Part II
- 07/21/2011: Final

© 2012, 101 companies
XML Programming (slides: .pdf, video: .mov)
Grammar Programming (slides: .pdf, video: .mov)
Generative Programming (slides: .pdf, video: .mov)
Intergalactic Programming (O/X) (slides: .pdf, video: .mov)
Intergalactic Programming (O/R) (slides: .pdf, video: .mov)
Database Programming (slides: .pdf, video: .mov)
Metaprogramming (slides: .pdf, video: .mov)
Aspect-oriented Programming (slides: .pdf, video: none)
Distributed Programming (slides: .pdf, video: .mov)
Pattern-oriented Programming (slides: .pdf, video: none)
Pattern-oriented Programming cont'd
Functional OO Programming (slides: .pdf, video: .mov)
Multithreaded programming (slides: .pdf, video: .mov)
Data-parallel Programming (slides: .pdf, video: none)
Q&A (slides: .pdf, video: none)
Course on modern web programming

This page makes available the slide material for a course on modern web programming. The slides rely on various 101 implementations which are linked accordingly.

Index of slide decks

- Introduction to the course: .pdf
- Web programming basics: .pdf
- Server-side scripting: .pdf
- REpresentational State Transfer: .pdf
- Model-View-Controller: .pdf
- JavaScript and AJAX: .pdf
- Cascading Style Sheets: .pdf
Course on modern data technologies

This page makes available the slide material for a course on modern data technologies.

The slides rely on various 101 implementations which are linked accordingly.

Index of slide decks

- Introduction to the course: .pdf
- Data technologies in action: .pdf
- MapReduce-style data processing: .pdf
- NoSQL -- What's that: .pdf
- Specific database systems:
  - Neo4J: .pdf
  - Riak: .pdf
  - MongoDB: .pdf
  - HBase: .pdf
- Use cases of NoSQL: .pdf
101 companies:

What about *abstraction*?
Abstraction opportunities

- **Classify** software technologies & languages.

- **Associate**
  - systems (101impls) w/ technologies & languages,
  - systems w/ features (capabilities, ...),
  - systems w/ software concepts,
  - languages & technologies w/ software concepts.

- **Model**
  - features of systems,
  - usage patterns of technologies,
  - usage instance in systems,
  - association rules.
Classification

[−] Language
  [+ ] 101language
  [+ ] Bytecode language
  [+ ] Domain-specific language
  [+ ] Format language
  [+ ] Markup language
  [+ ] Metadata language
  [+ ] Metalanguage
  [+ ] Modeling language
  [+ ] Programming language
  [+ ] Query language
  [+ ] Scripting language
  [+ ] Tool-defined language
  [+ ] Transformation language
  [+ ] XML language

[−] Technology
  [+ ] 101technology
  [+ ] Application technology
  [+ ] Comprehension technology
  [+ ] Development technology
  [+ ] Language technology
  [+ ] Mapping technology
  [+ ] Modeling technology
  [+ ] Office application
  [+ ] Package manager
  [+ ] Programming technology
  [+ ] Revision control tool
  [+ ] Standard
  [+ ] Web application
Associations for 101 implementation: jaxbComposition
Technologies

- Eclipse
- JAXB
- JUnit
- make

Features

- Serialization
- Tree structure
- Type-driven query
- Type-driven transformation
Associations for
101 implementation: **antlrObjects**
Features of 101 implementations

[-] 101feature
[+] Behavioral 101feature
[+] Meta-level 101feature
[+] Quality 101feature
[+] Structural 101feature
[+] UI 101feature

- Behaviors of the 101companies System
- Meta-level features
- Qualities of the 101companies System
- Structure of the 101companies System
- UI features of the 101companies System
[+] Behavioral 101feature
[+] Meta-level 101feature
[+] Quality 101feature
[+] Structural 101feature
[+] UI 101feature
[+] 101feature
[-] Behavioral 101feature
  Data export
  Data import
  Logging
  Structure-driven query
  Type-driven query
  Type-driven transformation
  Visualization
[+] Meta-level 101feature
[+] Quality 101feature
[+] Structural 101feature
[+] UI 101feature

- Behaviors of the 101companies System
- An export operation for company data
- An import operation for company data
- Logging of salary changes
- A query for the depth of department nesting
- A query for totaling the salaries of all employees
- A transformation for a salary cut
- The visualization of company data

- Meta-level features
- Qualities of the 101companies System
- Structure of the 101companies System
- UI features of the 101companies System
Model of YACC usage
Model O/X mapping with xsd.exe

http://softlang.uni-koblenz.de/mega/
Model a product using xsd.exe
The upper frame uses the MegaL/yEd visual notation for megamodeling. The lower frame shows some linked artifacts explained later in the paper.

Fig. 1. The linguistic architecture of a software product when displayed with the MegaL/Explorer tool. Technology for Object/XML mapping are clearly identifiable. Consider, for example, the fact that the class generator is not described as generating 'arbitrary' C\texttt{w}. Instead, a designated subset,\texttt{CSharpFromXsd}, is used because the generator indeed produces very regular code whose regularity helps with understanding Object/XML mapping, as we discuss later.

http://black42.uni-koblenz.de/production/101worker/MegaModels/implementations/xsdClasses/
Thanks!

• Questions?
• Comments?
• Please, get involved!