

Wikiplia:

The Free Programming Language That Anyone Can Edit

Tom Murphy VII

1 April 2007

Abstract

We present a new programming language called Wikiplia. The language has an unprecedented level of integration: The system is its own compiler, language definition, documentation, development environment, distributed filesystem, database, revision control system, bootstrapping software license, community message board, and World Wide Web home site. Wikiplia is designed to be Free to a greater extent and in more dimensions than existing languages.

Keywords: Freedom, programming language, software license, wiki, XML, weatherproof footwear and their fastening mechanisms, hyper-driven devices

1 Introduction

One of the most cherished social principles of mankind is freedom,¹ in its many incarnations. More recently, freedom has become an important principle in computer science as well, with the introduction of Free Software licenses such as the GNU GPL,² extensible markup languages such as XML,³ the ability explicitly deallocate memory with the `free(3)` library call, and the widespread availability of Free Herbal Viagra on the World Wide Web.⁴

The aim of this project is to develop a programming language that is as free as possible. We begin

by enumerating freedoms that we desire to support. Because freedom is a possession of inestimable value⁵ we do not attempt to rank these freedoms; instead, each freedom is “numbered” using a symbol drawn from incomparable sets of glyphs.

Freedom ©: The freedom to tinker. Users should be able to study a program to see how it works, and to make modifications to suit his or her needs. For most software, this means that the programmer needs access to the software’s documentation, source code, and UML⁶ use case diagrams. This is traditionally achieved through licenses such as the GPL; however, as we will discuss in Section 2 there are special considerations for bootstrapping compilers that render the GPL inadequate for this purpose.

Freedom \hbar : Freedom of expression. Programmers should be able to write their programs using any expressions that they like. Specifically, there should be no prior establishment of arbitrary categories of expression that are excluded, such as those that discriminate on the basis of class, mathematical philosophy, or type.

Freedom Ω : Free to be You and Me. The development of a programming language should not be confined to the bearded academic elite, gazing down upon the programmer fiefs from their stratospheric ivory towers. Wikiplia is the free programming language that *anyone* can edit: from bearded academic elite⁷ to congressional staffers⁸ to nameless Slashdot⁹ trolls. Wikiplia’s WWW-based approach means that

¹Copyright © 2007 the Regents of the Wikiplia Foundation. Appears in SIGBOVIK 2007 with the permission of the Association for Computational Heresy; *IEEEEE!* press, Verlag-Verlag volume no. 0x40-2A. This document may be distributed under the terms of the TIA Public License (Section 5). £0.00

²Wikipedia, the free encyclopedia: *Freedom*; 2007

³Wikipedia, the free encyclopedia: *GNU General Public License*; 2007

⁴Wikipedia, the free encyclopedia: *XML*; 2007

⁵Wikipedia, the free encyclopedia: *Sildenafil*; 2007

⁶Wikipedia, the free encyclopedia: *Cicero*; 2007

⁷Wikipedia, the free encyclopedia: *Unified Modeling Language*; 2007

⁸Wikipedia, the free encyclopedia: *Tenure*; 2007

⁹Wikipedia, the free encyclopedia: *Congressional staffer edits to Wikipedia*

¹⁰Wikipedia, the free encyclopedia: *Slashdot*; 2007

an Internet connection and compatible WWW hypertext browser is all that's needed to begin on the life-changing journey of programming language design.

Freedom Δ : Freedom of beer. Users should be able to write software without paying money to a licensing authority or certification program.

Freedom \P : Freedom to redefine freedom. Freedom should be free, so the definition of freedom should be free to change as the meaning of freedom changes. Wikiplia's license allows for Wikiplia to be distributed in a way that monotonically increases freedom as new concepts and catchphrases of freedom are invented.

Freedom $x^{1.2}\sqrt{60 + \frac{x}{z^2}}$: Freedom of USA #1. Wikiplia is 100% made in the USA and only available in English.¹⁰

2 Reflections on strapping straps and booting boots

The hallmark of Free software is the GNU General Public License. It is a hereditary license that requires that (a) source code be distributed with the program and (b) modified versions of the program also be licensed under the GPL. The intention is that anyone receiving the software can exercise Freedom \textcircled{C} by understanding the source code and modifying it to suit his needs. Clearly any source code will not do: an obfuscated¹¹ version of the source code cannot be easily understood or modified, even though it is technically "source code." The GPL therefore legally defines source code as the "preferred form" for "making modifications."

Even source code in the preferred form might not be enough to achieve Freedom \textcircled{C} , however. For instance, the program might be written in a mysterious programming language that only the author understands, and that programming language might only be implemented in a private compiler on the author's hard disk.¹² It is therefore reasonable to construe the "preferred form" of the original software to include the implementation of the programming language that the software is written in. Because the

programmer might need to fix bugs or extend the programming language implementation in order to modify the original program, he also needs the source code for that language as well. This code must also be written in some language, so the process continues. It can end when one of the programming languages is generally well known enough that there are no practical barriers to understanding it or finding an implementation (examples would include C¹³ and ALGOL 58¹⁴), or so simple that the implementation is essentially non-existent (*e.g.* an assembler implemented directly in machine code).

Another way for this process to terminate is for a programming language to be implemented in itself. This is known as a "bootstrapping compiler." A natural social tendency causes this to be very common: language implementors are more likely to enjoy the language they are implementing, and therefore more likely to choose it to implement the language. But when this process terminates this way, the reader might be left with a suspicious sense that nothing has actually been achieved.¹⁵ Specifically: What freedom-fulfilling use is the source code to an implementation of a mysterious programming language, if that source code is itself in the same mysterious programming language?

Let us concentrate on a more concrete example. The GNU C compiler¹⁶ (licensed under the GPL) is an implementation of the C language with some extensions specific to the compiler. The GCC source code uses some of these extensions. Can the GCC be Free software if it requires the GCC to build? In the extreme case, what if someone were to add an extension to the GCC to enable a new C keyword—called `compile_a_program`—and then replace the entire source code with:

```
int main (int argc, char ** argv) {
    compile_a_program;
    return 0;
}
```

Such code is clearly worthless. Not all subversions of the source code via language extension may be so overt, but we claim that they nonetheless pose a substantial threat to freedom.

We do not wish to limit the programmer's ability to make extensions to a language, since this would

¹⁰Wikipedia, the free encyclopedia: *Freedom fries*; 2007

¹¹Wikipedia, the free encyclopedia: *Obfuscated code*; 2007

¹²Wikipedia, the free encyclopedia: *Hard disk*; 2007

¹³Wikipedia, the free encyclopedia: *C (programming language)*; 2007

¹⁴Wikipedia, the free encyclopedia: *ALGOL 58*; 2007

¹⁵In the case of a LOGO interpreter implemented in LOGO, we could say that this is then "turtles all the way down."

¹⁶Wikipedia, the free encyclopedia: *GNU Compiler Collection*; 2007

also toe-step Freedom ©. We then conclude that the licensing terms must be expanded in order to provide more than “source code.” We propose that not only the source, but the source code’s *history*, must be made available.

2.1 Revision control

Computer scientists use revision control¹⁷ to track changes to software and to coordinate development between multiple programmers. This has been true for thousands of years. Popular revision control systems such as CVS¹⁸ and Subversion¹⁹ allow for code to be concurrently modified by two or more developers and then have their changes integrated after the fact by an explicit “check in” and conflict resolution phase.

It may naïvely seem that publishing the entire CVS history of a project would solve the issue with language extensions: By inspecting the revision that introduced the `compile_a_program` feature (but prior to the replacement of the GCC with the minuscule version above), one could then see its implementation and then know what it means. For certain patterns of development this does indeed suffice. However, programmers are not forced to check in their changes except at their own whims, as determined by social conventions; a programmer might make the private addition of the keyword `compile_a_program`, then rewrite the GCC to use it, and only then check in this change as one revision. For this action he will surely be rebuked by his fellow programmers; none of the other developers can compile the new version of the code without access to the intermediate revision! This social pressure would also naïvely seem to be enough to address the problem, but more insidious scenarios yet obtain.

As a concrete example, suppose there are two programmers called K and R. Each is modifying the GCC with the purpose of adding a new character constant, `'\c'`. K and R start at revision 100 of the GCC. K finds the case analysis for parsing character constants:

```
/* REVISION 100 (K) */
switch(ch) {
  case 'n': return '\n';
  case 'r': return '\r';
  :
  default: abort("bad char constant");
}
```

He adds a case for his extension, without using the extension, and checks this in as revision 101.

```
/* REVISION 101 (K) */
switch(ch) {
  case 'n': return '\n';
  case 'r': return '\r';
  :
  case 'c': return 257;
  default: abort("bad char constant");
}
```

Meanwhile, R has similar (but not identical) inspiration and modifies his copy of the compiler:

```
/* REVISION 100 (R) */
switch(ch) {
  case 'n': return '\n';
  case 'r': return '\r';
  :
  case 'c': return 8675309;
  default: abort("bad char constant");
}
```

He does not commit his code because he is wary of the time-consuming conflict resolution phase and is late for a date with K’s estranged wife who is fed up with K’s all-night hacking binges. He burns rubber in his 2007 Honda Civic²⁰ with aftermarket spoiler for a night on the town, believing that a healthy well-rounded programmer spends more or less equal nights basking in the pale amber glow of the teletype as waking up with a few missing teeth naked and norovirused in some midtown alleyway with his wallet barely out of reach but empty anyway, having amply exercised Freedom Δ .

Meanwhile, K continues extending the GCC, using the extension to implement itself. He checks in this code with no conflicts:

¹⁷Wikipedia, the free encyclopedia: *Revision control*; 2007

¹⁸Wikipedia, the free encyclopedia: *Concurrent Versions System*; 2007

¹⁹Wikipedia, the free encyclopedia: *Subversion (software)*; 2007

²⁰Wikipedia, the free encyclopedia: *Honda Civic*; 2007

```

/* REVISION 102 (K) */
switch(ch) {
  case 'n': return '\n';
  case 'r': return '\r';
  :
  case 'c': return '\c';
  default: abort("bad char constant");
}

```

K punches out at 1130 UTC,²¹ just as R returns from his adventure. R's confidence bolstered, he finishes his extension effort, following best practices and implementing the extension using itself:

```

/* REVISION 100 (R) */
switch(ch) {
  case 'n': return '\n';
  case 'r': return '\r';
  :
  case 'c': return '\c';
  default: abort("bad char constant");
}

```

He now decides to commit his changes (forgetting that he did not commit the intermediate revision). To do so he updates to the newest revision, 102, and sees that there are no conflicts—in fact, revision 102 already contains his changes! Believing that his changes are therefore compatible, he continues hacking.

After this scenario, K and R believe they are working on the same programming language—after all, it has the same source code—but their minor bifurcation in development history means that they have forever different meanings of the '\c' extension. This mistake is likely to go unnoticed for some time, and until it is resolved, the meaning of the '\c' extension is firmly enslaved in the bipartite penitentiary of double entendre, yearning to be free...²²

²¹Wikipedia, the free encyclopedia: *Coordinated Universal Time*; 2007

²²Wikipedia, the free encyclopedia: *Information wants to be free*; 2007

2.2 Solution

Based on these scenarios we conclude that extant social measures such as revision control conventions are not enough to guarantee freedom in all circumstances. Even if we think these situations are implausible in the hands of well-intentioned, well-mannered and capable^{23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45} software engineers, we wish for our software to remain free even when in the hands of nefarious and crafty factions who would seek to fracture⁴⁶ our free software community. We therefore need a technological and legal solution that forces the entire development history to be available. Keeping with Freedom $x^{1.2}\sqrt{oo + \frac{x}{z^2}}$, we call this technology and its license *Total Information Awareness* after the successful project of the US Government with the same name.⁴⁷

Technologically, we develop our system around a primitive notion of revision control in which every change to the system is recorded. Because the system has an integrated editor, every action of a programmer is logged and preserved indefinitely with no extra action necessary on the programmer's part. Such commits are globally atomic, using a single universal repository. (This means that in the above scenario, R would not have been able to forget to commit his intermediate change, and would have been forced to observe his conflict with K.) To protect against the possibility that divergent development paths lead to incompatible compilers, we require that there is only one compiler for any given programming language, which itself exists in the revision control system. Therefore, it is always clear which version of

²³Wikipedia, the free encyclopedia: *Year 2000 problem*; 2007

²⁴Wikipedia, the free encyclopedia: *Northeast Blackout of 2003*

²⁵Wikipedia, the free encyclopedia: *Ariane 5 Flight 501*; 2007

²⁶Wikipedia, the free encyclopedia: *Mars Climate Orbiter*; 2007

²⁷Wikipedia, the free encyclopedia: *Mars Polar Lander*; 2007

²⁸Wikipedia, the free encyclopedia: *Mars Rover*; 2007

²⁹Wikipedia, the free encyclopedia: *Mars Pathfinder*; 2007

³⁰Wikipedia, the free encyclopedia: *Gripen#Crashes*; 2007

³¹Wikipedia, the free encyclopedia: *missingno.*; 2007

³²Wikipedia, the free encyclopedia: *XSS*; 2007

³³Wikipedia, the free encyclopedia: *Buffer overflow*; 2007

³⁴Wikipedia, the free encyclopedia: *Therac-25*; 2007

³⁵Wikipedia, the free encyclopedia: *Lothar (storm)*; 2007

³⁶Wikipedia, the free encyclopedia: *Mariner 1*; 2007

³⁷Wikipedia, the free encyclopedia: *Code Red worm*; 2007

³⁸Wikipedia, the free encyclopedia: *SQLSlammer*; 2007

³⁹Wikipedia, the free encyclopedia: *Sandstorm (vehicle)*; 2007

⁴⁰Wikipedia, the free encyclopedia: *Samy (XSS)*; 2007

⁴¹Wikipedia, the free encyclopedia: *Pentium FDIV bug*; 2007

⁴²Wikipedia, the free encyclopedia: *MIM-103 Patriot*; 2007

⁴³Wikipedia, the free encyclopedia: *Windows 95*; 2007

⁴⁴Wikipedia, the free encyclopedia: *Morris (computer worm)*; 2007

⁴⁵Wikipedia, the free encyclopedia: *USS Yorktown (CG-48)*; 2007

⁴⁶Wikipedia, the free encyclopedia: *Fork (software development)*; 2007

⁴⁷Wikipedia, the free encyclopedia: *Information Awareness Office*; 2007

the compiler was used to produce an executable from source. This also makes the K and R scenario impossible: there is only one compiler and so it is impossible for it to differ from itself.

Legally, all of the software is licensed (Section 5) under terms similar to the GNU GPL, but that define the “source code” to include the entire revision history of the system. In order to be compatible with Freedom \blacklozenge , we allow the license itself to be edited, but to ensure that no one can remove freedoms already present in the license, the license includes a provision that allows any *prior* version of the license to be used, at the programmer’s option.

2.3 Implementation

Wikiplia, the free programming language that anyone can edit, is implemented as a web-site on the Internet at the address

`http://wikiplia.spacebar.org:2222/`

2.4 Roadmap

The remainder of this paper proceeds as follows. We first present in Section 3 the design of the initial Wikiplia system, which is used to bootstrap the rest of Wikiplia. We then discuss the current state of Wikiplia as of revision 532 in Section 4. We explain the freedom-preserving TIA Public License in Section 5. We conclude with a discussion of unrelated work and plans for the future 6.

3 Core calculus

$$X ::= \langle \text{tag} \rangle X_1 X_2 \dots X_n \langle / \text{tag} \rangle$$

| string

Figure 1: Syntax of XML

Wikiplia is built upon a core calculus of structured data with primitive revision control. Because we wish to support the freedom to tinker, the structured data take the form of XML (the *extensible* markup language; Figure 1). Similar to the W3C’s XML Validation,⁴⁸ we allow the quality of an XML document to be assessed via a process called *evaluation*, whose output (if any) is itself an XML document.

⁴⁸Wikipedia, the free encyclopedia: *XML schema*; 2007

Selected rules for XML evaluation (the dynamic semantic markup) are given in Figures 2 and 3.⁴⁹

Revision control is accessed through the class of imperative *cvs* judgments. We assume a single global repository, which maps *keys* (strings) to lists of *revisions*. A revision is a monotonically increasing and unique *revision number* (integer) paired with an XML document. The judgment *cvs commit s X = i* creates a new revision with revision number *i* and data *X* and inserts it under the key *s*. The judgment *cvs checkout s = X* fetches the most recent revision for the key *s*;⁵⁰ the document *X* (if no such key exists, then the document is invalid). The judgment *cvs checkout -r i s = X* does the same, but for the specific revision number *i*⁵¹ (if no such revision exists, the document is invalid). Finally, *cvs log s = X* fetches all of the revision numbers for the key *s*, as a series of integers \vec{X} .

3.1 Syntax

$$E ::= \begin{array}{l} (E_1 E_2 \dots E_n) \\ \text{"string"} \\ n \\ \text{symbol} \\ 'E \end{array}$$

$$\begin{array}{l} \llbracket (E_1 \dots E_n) \rrbracket = \langle \text{list} \rangle \llbracket E_1 \rrbracket \dots \llbracket E_n \rrbracket \langle / \text{list} \rangle \\ \llbracket "s" \rrbracket = \langle \text{string} \rangle s \langle / \text{string} \rangle \\ \llbracket n \rrbracket = \langle \text{int} \rangle n \langle / \text{int} \rangle \\ \llbracket \text{sym} \rrbracket = \langle \text{symbol} \rangle \text{sym} \langle / \text{symbol} \rangle \\ \llbracket 'E \rrbracket = \langle \text{quote} \rangle \llbracket E \rrbracket \langle / \text{quote} \rangle \end{array}$$

Figure 4: The syntax for the XESP syntax. The recursively defined $\llbracket \cdot \rrbracket$ operation converts an XESP expression into an XML document.

XML documents are universally parseable.⁵² However, they are difficult to write and read. Therefore, as usual⁵³ we create a new syntax by which humans can write and read documents and which the computer automatically parses and converts to the eas-

⁴⁹XML documents can be self-correcting through the use of the *handle* primitive, which detects an invalid document and proceeds along an alternative path. We omit the rules for this feature, which requires propagating invalid document status throughout evaluation and thus complicates the rules substantially.

⁵⁰Wikipedia, the free encyclopedia: *Dynamic scope*; 2007

⁵¹Wikipedia, the free encyclopedia: *Static scope*; 2007

⁵²Wikipedia, the free encyclopedia: *Parsing*; 2007

⁵³Wikipedia, the free encyclopedia: *RELAX NG*; 2007

$$\begin{array}{c}
\frac{}{\Gamma \vdash \text{eval} \langle \text{string} \rangle s \langle / \text{string} \rangle \mapsto \langle \text{string} \rangle s \langle / \text{string} \rangle} \qquad \frac{}{\Gamma \vdash \text{eval} \langle \text{quote} \rangle X \langle / \text{quote} \rangle \mapsto X} \\
\frac{}{\Gamma \vdash \text{eval} \langle \text{int} \rangle s \langle / \text{int} \rangle \mapsto \langle \text{int} \rangle s \langle / \text{int} \rangle} \qquad \frac{}{\Gamma \vdash \text{eval} \langle \text{prim} \rangle s \langle / \text{prim} \rangle \mapsto \langle \text{prim} \rangle s \langle / \text{prim} \rangle} \\
\frac{s \text{ prim}}{\Gamma \vdash \text{eval} \langle \text{symbol} \rangle s \langle / \text{symbol} \rangle \mapsto \langle \text{prim} \rangle s \langle / \text{prim} \rangle} \qquad \frac{\Gamma(s) = X}{\Gamma \vdash \text{eval} \langle \text{symbol} \rangle s \langle / \text{symbol} \rangle \mapsto X} \\
\frac{}{\Gamma \vdash \text{eval} \langle \text{closure} \rangle \Gamma s X \langle / \text{closure} \rangle \mapsto \langle \text{closure} \rangle \Gamma s X \langle / \text{closure} \rangle} \\
\frac{\Gamma \vdash \text{eval} X_1 \mapsto X'_1 \quad \dots \quad \Gamma \vdash \text{eval} X_n \mapsto X'_n \quad \Gamma \vdash \text{rate} X'_1 \dots X'_n \mapsto X'}{\Gamma \vdash \text{eval} \langle \text{list} \rangle X_1 \dots X_n \langle / \text{list} \rangle \mapsto X'}
\end{array}$$

Figure 2: Evaluation of XML, part 1. The judgment $\Gamma \vdash \text{eval } X \mapsto X'$ indicates an assessment of the document X with value X' . The judgment rate is an auxiliary assessment of a list of documents. It is defined in Figure 3. \vec{X} is shorthand for a possibly empty sequence of XML documents. Γ is itself an XML document of the form $\langle \text{list} \rangle \langle \text{symbol} \rangle s_1 \langle / \text{symbol} \rangle X_1 \dots \langle \text{symbol} \rangle s_n \langle / \text{symbol} \rangle X_n \langle / \text{list} \rangle$. We take the judgment $\Gamma(s) = X$ to produce the leftmost X_i in Γ such that s_i is s . $\Gamma, s = X$ is $\langle \text{list} \rangle \langle \text{symbol} \rangle s \langle / \text{symbol} \rangle X \vec{X} \langle / \text{list} \rangle$ if Γ is $\langle \text{list} \rangle \vec{X} \langle / \text{list} \rangle$. The judgment $s \text{ prim}$ holds when s is one of insert, head, read, abort, lambda, list, cons, quote, string, xcase, size, sub, substr, handle, parse, eval, eq, +, -, int, history, let, or if.

ily parseable XML syntax and back to the new syntax, reducing complexity.⁵⁴ This compact syntax is based upon parentheses rather than tags: The XML document $\langle \text{list} \rangle X_1 X_2 \langle / \text{list} \rangle$ is instead written $(X_1 X_2)$. Note that the closing parenthesis is not *named* as in XML, which makes parsing difficult because the computer must *guess* which parenthesis belongs to which other parenthesis. We therefore call this compact syntax XESP because it basically *reads the programmer's mind*⁵⁵ to guess what the name of the closing parenthesis should be. The grammar for XESP is given in Figure 4 along with the translation to XML documents. From now on, we use the XESP syntax in our examples.

3.2 Implementation

Wikiplia is implemented as a World Wide Web Home-Site, which allows for easy access from any location.

The system is implemented as a Standard ML⁵⁶ program of approximately 1,000 lines.⁵⁷ This program

⁵⁴For efficiency, the Wikiplia implementation is optimized to lazily perform these translations, so that the document is never represented in XML form.

⁵⁵Wikipedia, the free encyclopedia: *Extra-sensory perception*; 2007

⁵⁶Wikipedia, the free encyclopedia: *Standard ML*; 2007

⁵⁷This count does not include general purpose libraries, such as a networking library.

is designed to be minimal: it consists of a web server, a revision control system, and facilities for evaluating XESP documents. It also contains a very minimal bootstrapping “compiler” for XESP documents, with its boots manually strapped. From this tiny core we then develop the remainder of Wikiplia using Wikiplia itself.

Some may balk at the choice of Standard ML, as the language is miserably non-free: First, while many of the major implementations are GNU or BSD-licensed,⁵⁸ all are implemented in Standard ML itself, yielding the bootstrapping problem described earlier. Second, the mathematical Definition of Standard ML book is only available as a copyrighted publication of MIT Press,⁵⁹ not even *ostensibly* in a free manner. However, the performance considerations of the server and evaluator—and the lack of suitable free alternatives—force us to settle for such subjugation.

3.2.1 Web Server

The web server’s job is simple. It runs in a loop, accepting a single connection, setting up an initial environment Γ for evaluation (which associates the symbols `request.url`, `request.ip` and `request.time` with

⁵⁸Wikipedia, the free encyclopedia: *BSD licenses*; 2007

⁵⁹Wikipedia, the free encyclopedia: *MIT Press*; 2007

$$\begin{array}{c}
\frac{}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{list} \langle / \text{prim} \rangle \vec{X} \mapsto \langle \text{list} \rangle \vec{X} \langle / \text{list} \rangle} \\
\frac{}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{cons} \langle / \text{prim} \rangle X \langle \text{list} \rangle \vec{X} \langle / \text{list} \rangle \mapsto \langle \text{list} \rangle X \vec{X} \langle / \text{list} \rangle} \\
\frac{}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{lambda} \langle / \text{prim} \rangle \langle \text{symbol} \rangle s \langle / \text{symbol} \rangle X \mapsto \langle \text{closure} \rangle \Gamma s X \langle / \text{closure} \rangle} \\
\frac{\Gamma', s = \langle \text{list} \rangle \vec{X} \langle / \text{list} \rangle \vdash \text{eval} X \mapsto X'}{\Gamma \vdash \text{rate} \langle \text{closure} \rangle \Gamma' s X \langle / \text{closure} \rangle \vec{X} \mapsto X'} \quad \frac{\Gamma \vdash \text{eval} X \mapsto X'}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{eval} \langle / \text{prim} \rangle X \mapsto X'} \\
\frac{\Gamma \vdash \text{eval} X \mapsto X'}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{xcase} \langle / \text{prim} \rangle \langle \text{list} \rangle \langle / \text{list} \rangle X \vec{X} \mapsto X'} \\
\frac{\Gamma, s_h = X_h, s_t = X_t \vdash \text{eval} X_b \mapsto X'}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{xcase} \langle / \text{prim} \rangle \langle \text{list} \rangle X_h \vec{X}_t \langle / \text{list} \rangle X_0 \langle \text{list} \rangle \langle \text{symbol} \rangle s_h \langle / \text{symbol} \rangle \langle \text{symbol} \rangle s_t \langle / \text{symbol} \rangle X_b \langle / \text{list} \rangle \vec{X} \mapsto X'} \\
\frac{\Gamma, s_q = X \vdash \text{eval} X_b \mapsto X'}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{xcase} \langle / \text{prim} \rangle \langle \text{quote} \rangle X \langle / \text{quote} \rangle X_0 X_1 \langle \text{list} \rangle \langle \text{symbol} \rangle s_q \langle / \text{symbol} \rangle X_b \langle / \text{list} \rangle \vec{X} \mapsto X'} \\
\frac{\Gamma \vdash \text{eval} X \mapsto X'}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{xcase} \langle / \text{prim} \rangle \langle \text{string} \rangle s \langle / \text{string} \rangle X_0 X_1 X_2 X \vec{X} \mapsto X'} \\
\frac{\Gamma \vdash \text{eval} X \mapsto X'}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{xcase} \langle / \text{prim} \rangle \langle \text{int} \rangle i \langle / \text{int} \rangle X_0 X_1 X_2 X_3 X \vec{X} \mapsto X'} \\
\frac{\Gamma, s_b = \langle \text{string} \rangle s \langle / \text{string} \rangle \vdash \text{eval} X_b \mapsto X'}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{xcase} \langle / \text{prim} \rangle \langle \text{symbol} \rangle s \langle / \text{symbol} \rangle X_0 X_1 X_2 X_3 X_4 \langle \text{list} \rangle \langle \text{symbol} \rangle s_b \langle / \text{symbol} \rangle X_b \langle / \text{list} \rangle \vec{X} \mapsto X'} \\
\frac{t = \text{prim or closure} \quad \Gamma \vdash \text{eval} X \mapsto X'}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{xcase} \langle / \text{prim} \rangle \langle \text{t} \rangle \vec{X}_t \langle / \text{t} \rangle X_0 X_1 X_2 X_3 X_4 X_5 X \vec{X} \mapsto X'} \\
\frac{}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{quote} \langle / \text{prim} \rangle X \mapsto \langle \text{list} \rangle X \langle / \text{list} \rangle} \\
\frac{\text{cvs checkout } s = X}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{head} \langle / \text{prim} \rangle \langle \text{string} \rangle s \langle / \text{string} \rangle \mapsto X} \\
\frac{\text{cvs checkout } -r i s = X}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{read} \langle / \text{prim} \rangle \langle \text{string} \rangle s \langle / \text{string} \rangle \langle \text{int} \rangle i \langle / \text{int} \rangle \mapsto X} \\
\frac{\text{cvs commit } s X = i}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{insert} \langle / \text{prim} \rangle \langle \text{string} \rangle s \langle / \text{string} \rangle X \mapsto \langle \text{int} \rangle i \langle / \text{int} \rangle} \\
\frac{\text{cvs log } s = \vec{X}}{\Gamma \vdash \text{rate} \langle \text{prim} \rangle \text{history} \langle / \text{prim} \rangle \langle \text{string} \rangle s \langle / \text{string} \rangle \mapsto \langle \text{list} \rangle \vec{X} \langle / \text{list} \rangle}
\end{array}$$

Figure 3: Evaluation of XML, part 2. The rate judgment assesses a sequence of XML expressions. The rules for rating the primitives parse, string, sub, substr, +, -, int, eq and if are omitted for space.

the appropriate values), and then evaluating the document contained at the head of the key `main` in the repository. The result of that evaluation is sent back to the web browser as a string.⁶⁰ The server knows nothing else about how Wikiplia works.

The web server is single-threaded (so each request must finish before the next is handled) because we desire global atomicity (Section 2).

3.2.2 Revision Control

The revision control system stores XESP documents and their history. This is mostly a straightforward implementation of the imperative `cvs` judgments given in Section 3. Unlike typical revision control systems, we need to support very large numbers⁶¹ of revisions with small edits (since every change is saved), so the implementation is engineered to make storing small revisions very cheap. Particularly, revisions are aggressively compressed by only storing the newest revision directly, and then a series of difference “plans” that describe how to get the next older revision from the current one. We compute optimal plans using an efficient minimal edit distance⁶² calculation at the token⁶³ level. As of revision 532, the database is only 250 kilobytes.⁶⁴

3.2.3 Document Evaluation

The Wikiplia implementation also has facilities for parsing and evaluating XESP documents. These are a direct implementation of the evaluation rules in Figures 2 and 3.

3.2.4 Bootstrapping

The key `main` of Wikiplia is responsible for decomposing the URL and acting upon it however is appropriate. The goal of the bootstrapping process is to make Wikiplia self-sufficient⁶⁵ in the sense that the language can be edited from the web site implemented by `main`. To do so, the web site needs to be able to present the user with an edit box containing the current source of the `main` key and the ability to save his code into the database, overwriting the `main` key, in order to add functionality.

⁶⁰The HTTP result code may be modified if, for example, the resulting document is a HTTP redirect to another URL.

⁶¹The implementation also supports revision numbers of arbitrary size.

⁶²Wikipedia, the free encyclopedia: *Levenshtein distance*; 2007

⁶³Wikipedia, the free encyclopedia: *Token (parser)*; 2007

⁶⁴Wikipedia, the free encyclopedia: *Kilobyte*; 2007

⁶⁵Wikipedia, the free encyclopedia: *Self-sufficiency*; 2007

```
(lambda 's '(parse (xcase s 'no '(h _ h))))
```

Figure 5: The initial bootstrapping compiler.

The initial implementation of `main` provides for the simple ability to edit, save, and view the current version of keys in the repository. These three actions are encoded as the URLs `/edit/key`, `/save/key`, and `/view/_/key`. The view action is straightforward. The edit action displays an HTML textarea⁶⁶ containing the current value of the key and a button that submits the changes to the `save` url. The save action is the most complex. First, the submitted document is saved as the new version of the key. Then, if the key is of the form `base.extension`, the database is checked to see if there is a key called `extension:compile`. If so, the XESP document that is there is applied to the input document (a string) to produce a document that is saved at the key `base`. This allows us to develop languages that are automatically compiled when saved.

This initial functionality is implemented directly in the XESP language, whose extension is `b`; the bootstrapping “compiler” is just the built-in parser (Figure 5).

Arranging that the repository contain the correct keys to make this work is subtle. A small initialization phase sets up:

- `b:compile.b` The source code (a string) in Figure 5
- `b:compile` The parsed document corresponding to the above, such that `b:compile` applied to `b:compile.b` yields `b:compile`
- `main.b` The source code of the original “main” program
- `main` The parsed document corresponding to the above, such that `b:compile` applied to `main.b` yields `main`.

We do this by *anticipating*, during the initialization process, the *meaning* of `b:compile` so that we can perform that action (parsing) on `b:compile.b` to produce `b:compile`. This can only be achieved by fiat and this is the essence of bootstrapping.

After this minimal initialization, we can then exclusively use the web interface to develop and extend Wikiplia.

⁶⁶Wikipedia, the free encyclopedia: *Text box*; 2007

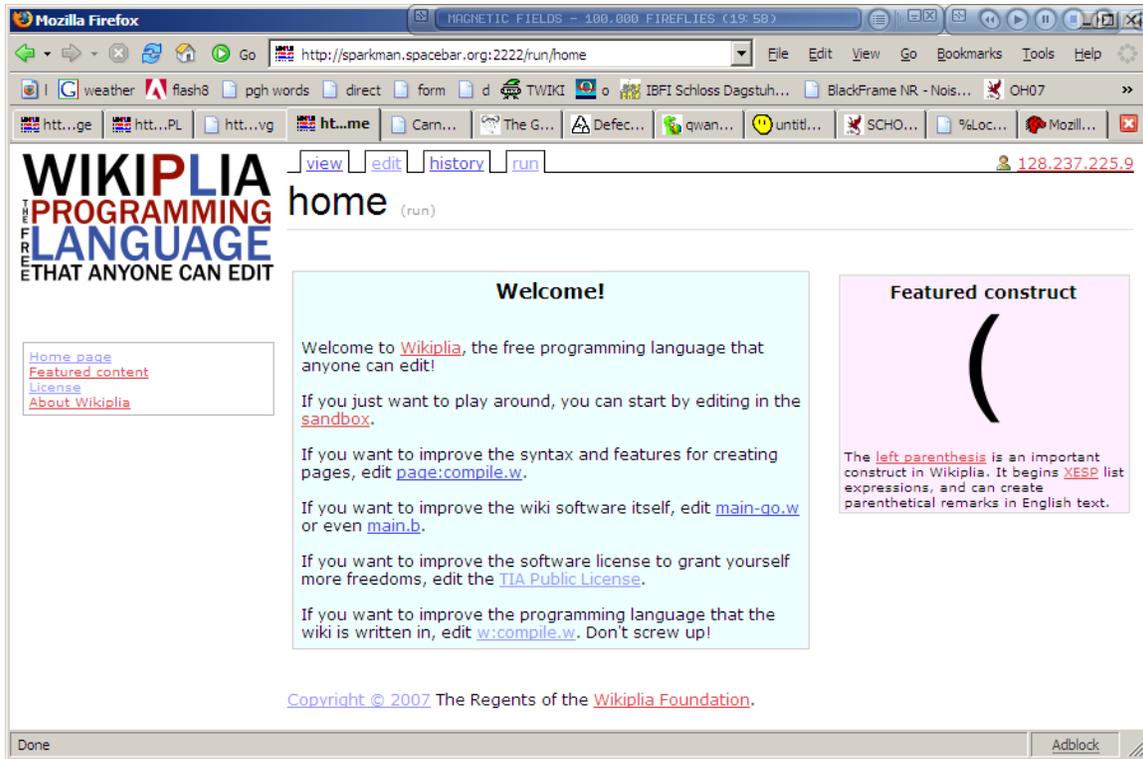


Figure 6: Screenshot of the Wikiplia home page as of revision 532.

4 Revision 532

As of writing, Wikiplia is at revision 532, and has a number of features implemented.

4.1 Interface

The editing interface implemented has been enhanced greatly; Figure 6 shows a screenshot of the main page. The various views of a key are shown with a series of tabs at the top of each page. Each user has a home page named after his IP address,⁶⁷ which he can use to catalogue his interests. Various warnings help the user, for example, if he tries to edit a page that was generated by compiling some source code, a warning message suggests that he may wish to edit the source code instead. An ornate logo in SVG⁶⁸ adorns the page, and a sidebar provides quick access to the site's features. The logo and graphics for the site are stored in the repository; the new `raw` and `typed` actions allow access to these resources over HTTP

so that they may be freely modified.⁶⁹ At revision 228 support for metadata was added for each page; a `history` tab now shows the date, revision number, IP address, and edit summary for each change to a key (Figure 7).

It is very easy to make mistakes that render the system unusable. Therefore Wikiplia supports the ability to safely revert to a previous version of a key. When making changes to the `main` key this ability can be accidentally disabled, so the complex functionality of `main` was split off to a new key called `main-go` at revision 9; the `main` key now only dispatches to `main-go` but provides an `emergency-revert` action that automatically reverts `main-go` to its previous revision in case it is damaged and the site is unusable.

⁶⁹Because Wikiplia is dogmatically forward-looking, the graphics require SVG support in the browser and support for the `data:` URL format, a combination only found in the newest versions of the Mozilla Firefox. Wikiplia is compatible with incompatible browsers, however, simply displaying a crapified version of each page.

⁶⁷Wikipedia, the free encyclopedia: *IP address*; 2007

⁶⁸Wikipedia, the free encyclopedia: *Scalable Vector Graphics*; 2007

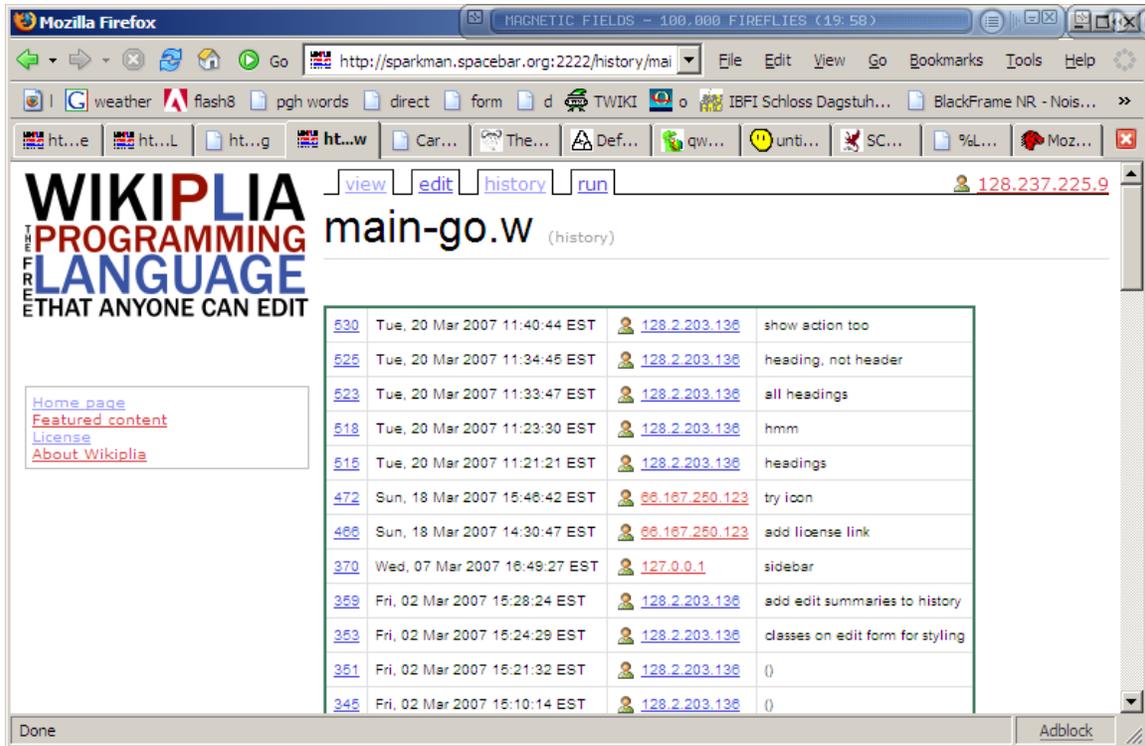


Figure 7: Screenshot of the history information for main-go.w.

4.2 Wikiplia language w

Writing XESP documents by hand is very tedious, so one of the first orders of business was to develop a compiler for a new language, `w`, which can be extended with convenient features. The first version of this language, created in revision 22, was written in XESP. It automatically quoted the appropriate arguments to the `let`, `it`, `lambda`, `xcase` primitives. In revision 361, the compiler was ported to the language `w` as `w:compile.w`, then compiled with the existing `b`-compiled version of `w:compile.b`, then recompiled with itself until reaching a fixed point.⁷⁰ After that, more features were added: a multi-argument function construct `fn`; a list-deconstructing binding construct `lets`; simple support for separately-compiled libraries⁷¹ via `include`; support for mutually-recursive⁷² bundles of functions via `fun`;⁷³ and the `cond` keyword for series of chained “if...else” conditionals. At each stage, the

⁷⁰Wikipedia, the free encyclopedia: *Fixed point (mathematics)*; 2007

⁷¹Wikipedia, the free encyclopedia: *Library (computing)*; 2007

⁷²Wikipedia, the free encyclopedia: *Recursion (computer science)*; 2007

⁷³Prior to this, recursion had to be encoded directly by passing an initial “self” argument to each function.

feature is implemented using the current version of `w`, and then `w:compile.w` is rewritten to use that convenient extension, and then recompiled until reaching a fixed point.

4.3 Wikiplia language page

Wikiplia is not just a programming language; it needs facilities for editing pages that are human readable as well. This can be used to edit documentation for programming languages, to modify the home page to tout new developments, to modify the software license (Section 5) or to deface other user’s personal pages. For this we provide a Wiki-like⁷⁴ syntax that allows for the authorship of such pages and easy linking between them. Like other Wikis, a link to a page that does not exist is colored red, to alert the user to the opportunity to stake out cyberspace real estate. This syntax is compiled to XESP documents via the `page` language; the resulting documents are active in the sense that they check the status of linked pages to report the correct color on every page load.

⁷⁴Wikipedia, the free encyclopedia: *Wiki*; 2007

5 TIA Public License

In this section we reproduce the Total Information Awareness Public License. Commentary is given via footnotes⁷⁵ into parts of the license.

TIA PUBLIC LICENSE
Revision 468, March 2007

BEGIN INVINCIBLE SECTION 1

This software is Copyright © 2007–∞
The Regents of the Wikiplia Foundation.
Permission is not granted to reproduce this
software or license except by the terms
explicitly enumerated below.

I. Invincible Sections

This license contains certain invincible
sections, denoted by the text “BEGIN
INVINCIBLE SECTION <n>” and “END
INVINCIBLE SECTION <n>”. Such
sections may not be modified under any
circumstances.

END INVINCIBLE SECTION 1⁷⁶

BEGIN INVINCIBLE SECTION 2

II. Version Identification and Invalid Licenses

This license must identify itself in the
header as Revision <n> for some number
<n>, which must be the same as the revision
number in the Wikiplia repository for the
key “TPL” in which the license text is
stored. If this is not the case, then
this version of the license is considered
Invalid and Void.

Permission is not granted to distribute
the software or license using any Invalid
version of the license.

END INVINCIBLE SECTION 2⁷⁷

⁷⁵Wikipedia, the free encyclopedia: *footnote*; 2007

⁷⁶Invincible sections exist in order to ensure the sanctity of the license. We first establish that invincible sections will appear and that they are inviolable; this itself is done in an invincible section. Invincible sections cause a limited loss of liberty, but this is the cost of freedom.

⁷⁷This invincible section establishes a connection between license versions and the actual contents of the Wikiplia repos-

BEGIN INVINCIBLE SECTION 3

III. Option of License

The licensee has the option to choose
any revision of the license prior to
(numerically less than) this version as
the licensing terms for the software and
license.

END INVINCIBLE SECTION 3

BEGIN INVINCIBLE SECTION 4

IV. Heredity of License

Any copy or derivative work of this
software or license must be licensed under
the TIA Public License.

END INVINCIBLE SECTION 4⁷⁸

BEGIN INVINCIBLE SECTION 5

V. Completeness of Copy

This software and license may only be
copied in their entirety, including the
entire revision history.

END INVINCIBLE SECTION 5⁷⁹

VI. Freedom to edit

Permission is hereby granted to edit this
license.⁸⁰

itory. Note the self-reference: Though this text is in an invincible section and never changes, the referent of “this license” does change as the rest of the license is modified. Because Wikiplia assigns version numbers monotonically, this ensures that the next invincible section is able to guarantee that freedom is monotonic. In the case of an invalid license, no permissions whatsoever are granted, so the licensee *must* use a prior valid version of the license. The initial version of the license is valid.

⁷⁸This clause makes the license “viral” like the GNU GPL, so that freedom is preserved in all descendants of the software.

⁷⁹This section is the centerpiece of Wikiplia; it guarantees that the “source code” to any software or programming language derived from Wikiplia is free from the loopholes described in Section 2 and so maximizes Freedom ©. Note that this does not limit the way that the software can be modified; the programmer might begin by blanking all of the keys he doesn’t care about—as long as he preserves the fact that there was *once* something there.

⁸⁰The only non-invincible provision of the original license allows the reader to add provisions that he desires to the license. This guarantees Freedom ¶, the freedom to redefine freedom.

This license is bootstrapping in the sense that it grants only the minimal permissions necessary, after setting up invariants via the behavior-limiting invincible sections. In fact, the original version of the license does not directly permit the licensee to copy the software at all; he must first amend the license using VI to give himself this permission.

6 Conclusion

We have reached the end of our journey. But the journey is not complete! We conclude with a discussion of future plans and unrelated work, and then conclude with another paragraph.

6.1 Future Work

Though Wikiplia in its current form is a usable general-purpose programming language, work remains to be done for it to reach its full potential. For one, it needs a vibrant community of contentious and hubristic editors hiding behind anonymous IP addresses boldly asserting half-baked syntactic extensions or enforcing superficial style preferences, gritting their teeth while typing and clicking white knuckled in a kind of road rage⁸¹ created by the dehumanizing semantic markup by which they are forced to communicate.

We also seek to improve the languages. The language `w` needs many more features to speed development: the parenthesis-based XESP syntax should eventually be replaced by a pleasant concrete syntax, if we can get around to it before too much code is written in XESP. A type system⁸² is not planned, because type systems restrict Freedom \hbar , the freedom of expression. However, we should seek to make Wikiplia as multi-paradigm as possible (again, freedom from discrimination on the basis of paradigm⁸³ orientation), supporting OOPs-oriented programming,⁸⁴ aspect-oriented programming,⁸⁵ duck-oriented typing,⁸⁶ orientation-oriented orienteering,⁸⁷ Orient-oriented

Note that even if a freedom-hater removes this provision from the license, the invincible sections above ensure that the freedom to edit the license is preserved for all time.

⁸¹Wikipedia, the free encyclopedia: *List of rages*; 2007

⁸²Wikipedia, the free encyclopedia: *Type system*; 2007

⁸³Wikipedia, the free encyclopedia: *paradigm*; 2007

⁸⁴Wikipedia, the free encyclopedia: *Object-oriented programming*; 2007

⁸⁵Wikipedia, the free encyclopedia: *Aspect-oriented programming*; 2007

⁸⁶Wikipedia, the free encyclopedia: *Duck typing*; 2007

⁸⁷Wikipedia, the free encyclopedia: *Orienteering*; 2007

programming,⁸⁸ etc.

The `page` language needs extensions for developing human-readable web pages, mostly for the purpose of creating jazzy graphics and boxes that distract from or directly call attention to obvious problems with the pages without actually addressing those problems.

6.2 Unrelated Work

All popular modern languages are defined via a definitional interpreter^{89,90,91} with accompanying O'Reilly “animal” book.⁹² The work on Wikiplia is unrelated: We have no animal mascot⁹³ and the languages are described by a tower of source-to-source translations⁹⁴ on top of a universally parseable semantic document in XML⁹⁵ form.

The author⁹⁶ doesn't think⁹⁷ much of musicals;⁹⁸ to be perfectly⁹⁹ honest,¹⁰⁰ so those are basically¹⁰¹ a no-go. He also feels that the metric system¹⁰² but paradoxically¹⁰³ also time zones¹⁰⁴ are pretty over-rated. Ketchup¹⁰⁵ on eggs¹⁰⁶ is gross,¹⁰⁷ but not quite as gross as foie gras¹⁰⁸, which more or less has the word¹⁰⁹ “gross” in its name¹¹⁰ so duh.¹¹¹

6.3 Another Paragraph

We have described Wikiplia, the free programming language that anyone can edit. Unlike other programming languages, it is designed to support a variety of freedoms (\copyright , \hbar , Δ , $x^{1.2}\sqrt{\infty} + \frac{x}{z^2}$, \P , \mathbb{R}) and is explicitly scalable to new freedoms. Wikiplia is implemented in a minimal bootstrapping core based on freedom-aware technologies such as XML, and then built up to a featured system using its own faculties

⁸⁸Wikipedia, the free encyclopedia: *The Orient*; 2007

⁸⁹Wikipedia, the free encyclopedia: *JavaScript*; 2007

⁹⁰Wikipedia, the free encyclopedia: *Objective Caml*; 2007

⁹¹Wikipedia, the free encyclopedia: *Perl*; 2007

⁹²Wikipedia, the free encyclopedia: *O'Reilly Media*; 2007

⁹³Wikipedia, the free encyclopedia: *ORLY owl*; 2007

⁹⁴Wikipedia, the free encyclopedia: *Translation*; 2007

⁹⁵Wikipedia, the free encyclopedia: *Category:ML programming language family*

⁹⁶Wikipedia, the free encyclopedia: *Author*; 2007

⁹⁷Wikipedia, the free encyclopedia: *Thought*; 2007

⁹⁸Wikipedia, the free encyclopedia: *Musical theatre*; 2007

⁹⁹Wikipedia, the free encyclopedia: *Thomas Aquinas*; 2007

¹⁰⁰Wikipedia, the free encyclopedia: *Honesty*; 2007

¹⁰¹Wikipedia, the free encyclopedia: *BASIC*; 2007

¹⁰²Wikipedia, the free encyclopedia: *Metric system*; 2007

¹⁰³Wikipedia, the free encyclopedia: *Pardaox*; 2007

¹⁰⁴Wikipedia, the free encyclopedia: *Time zone*; 2007

¹⁰⁵Wikipedia, the free encyclopedia: *Ketchup*; 2007

¹⁰⁶Wikipedia, the free encyclopedia: *Egg (food)*; 2007

¹⁰⁷Wikipedia, the free encyclopedia: *Gross*; 2007

¹⁰⁸Wikipedia, the free encyclopedia: *Foie gras*; 2007

¹⁰⁹Wikipedia, the free encyclopedia: *Word*; 2007

¹¹⁰Wikipedia, the free encyclopedia: *Name*; 2007

¹¹¹Wikipedia, the free encyclopedia: *Duh*; 2007

for extension. However, much work remains to be done. We invite you to join us!

<http://wikiplia.spacebar.org:2222/>