

Ruby

Extended Library Howto

arton

also not so known as Akio Tajima

arton (Akio Rubyist Tajima ONline)

ActiveScriptRuby

(Windows Installer Package with Ruby ActiveScript Interface)

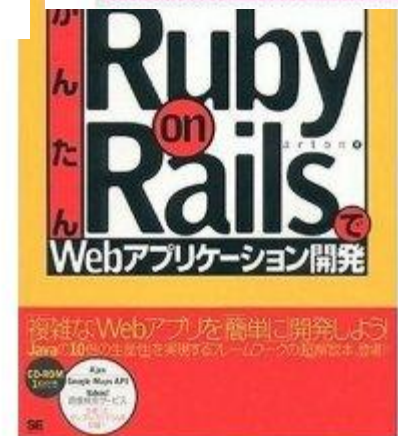
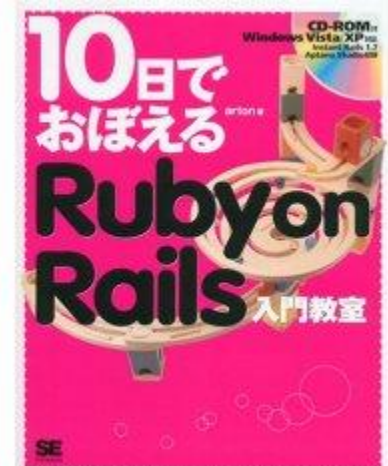
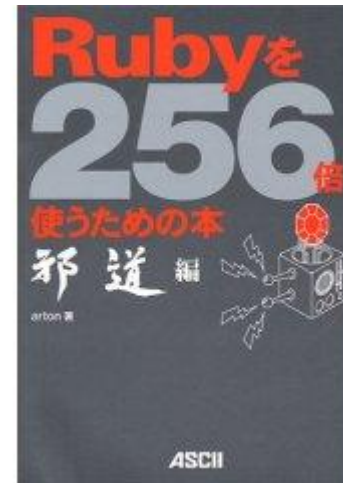
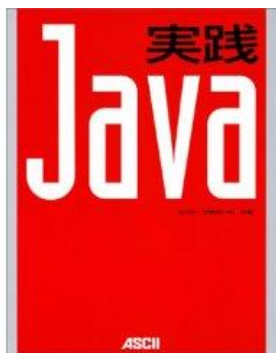
<http://arton.hp.infoseek.co.jp/>

Rjb(Ruby Java Bridge)

<http://rjb.rubyforge.org>

diary

<http://arton.no-ip.info/diary>

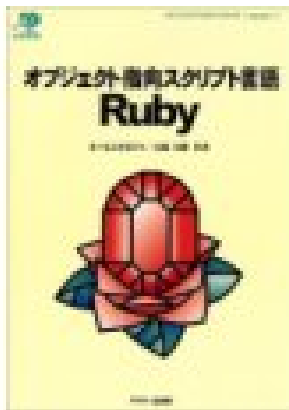


Agenda

- Ruby拡張ライブラリとは？
- 何が嬉しいの？
- 情報源
- 簡単な拡張ライブラリのデモ
- 特徴を生かすには？
- それなりの拡張ライブラリのデモ
- 代替案
- まとめ
- Q&A

拡張ライブラリ

- 拡張ライブラリとはCまたはC++言語で記述されていて、Rubyに組み込むことのできるライブラリです。



オブジェクト指向スクリプト言語Ruby
P.428

何が嬉しいの？

- 拡張ライブラリによる部品を上手に設計し、重い処理の本質的な部分をそれらの部品に任せることができれば、コンパイラ型言語によるプログラムと大差ない実行速度と、比較にならない開発効率を両立させることも可能です。

高速化の例

```
1.upto(10) { |x| puts x }
```

```
for i in 1..10; puts i; end
```

```
for (i = 0; i < 10; i++) {  
  rb_funcall(rb_stdout,  
             rb_intern("puts"),  
             1, INT2FIX(i));  
}
```

情報源

- README.EXT、README.EXT.ja (1995)
- オブジェクト指向スクリプト言語Ruby (1999)
- <http://www.loveruby.net/w/RubyExtensionProgrammingGuide.html>

- RHG



一次情報

- ruby.h
- intern.h
 - 主要クラスの公開API
- rubyio.h
 - (多分)
- rubysig.h
 - スレッド操作
- version.h
- st.h
 - ハッシュ

要求される技術

- ポインタ
 - 全部ポインタ
- ソースコードリーディング
 - Rubyのソースを読む
- セキュアコーディング



配布

- setup.rb
 - <http://www.loveruby.net/archive/setup/>
- Gem
 - gem & rake
- Binary
 - MSI,apt,dmg

デモ

- extrails
 - <http://svn.arton.no-ip.info/raa/trunk/extrails>
 - Project generation
 - extrails HelloWorld
 - » with Rakefile for creating Gem
 - Skeleton generation
 - ruby script/generate scaffold Hello say_hello say_bye
 - » with sample code for defining variable, calling method

拡張ライブラリの決まり

- void Init_ファイル名()
 - モジュール、クラスの登録
 - 定数、グローバル変数の作成

定数、グローバル変数

- 定数

```
rb_define_const(モジュール, “名前”, VALUE);
```

```
rb_define_global_const(“名前”, VALUE);
```

- グローバル変数

```
rb_define_variable(“名前”, VALUE*);
```

(GC防御の基本:

cf. `rb_gc_register_address(VALUE*)`)

```
rb_define_virtual_variable(“名前”, getter, setter);
```

モジュール、クラス、メソッド

```
モジュール = rb_define_module("名前");  
rb_define_module_function(モジュール, "名前",  
                           メソッド, arity);
```

項数情報:

0 無引数
-1 可変引数

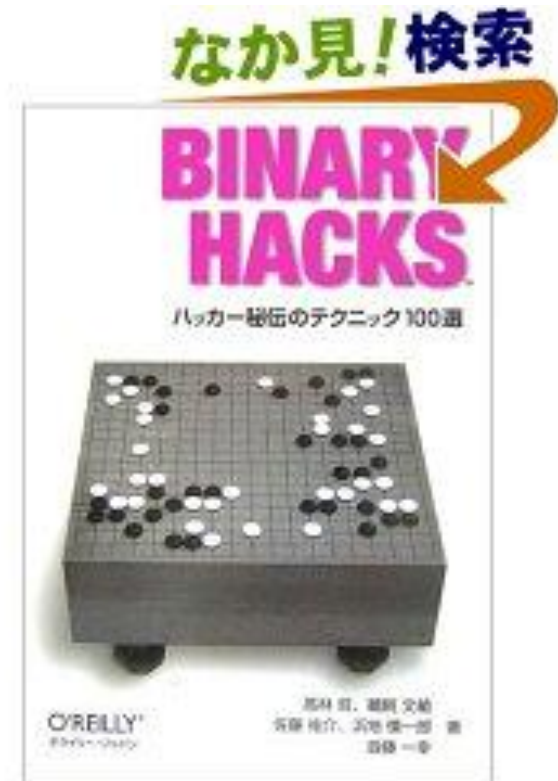
```
クラス = rb_define_class("名前", super);  
クラス = rb_define_class_under(モジュール, "名前", super);  
rb_define_method(クラス, "名前", メソッド, arity);
```

特徴を生かす

- × テキスト処理
- × ネットワーク処理
- × ファイル操作

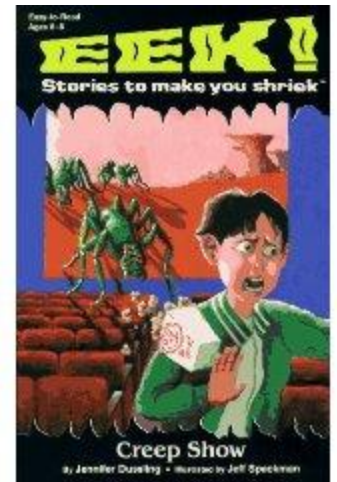
特徴を生かす

- アドレス操作
- 割り込み処理
- ネイティブAPI

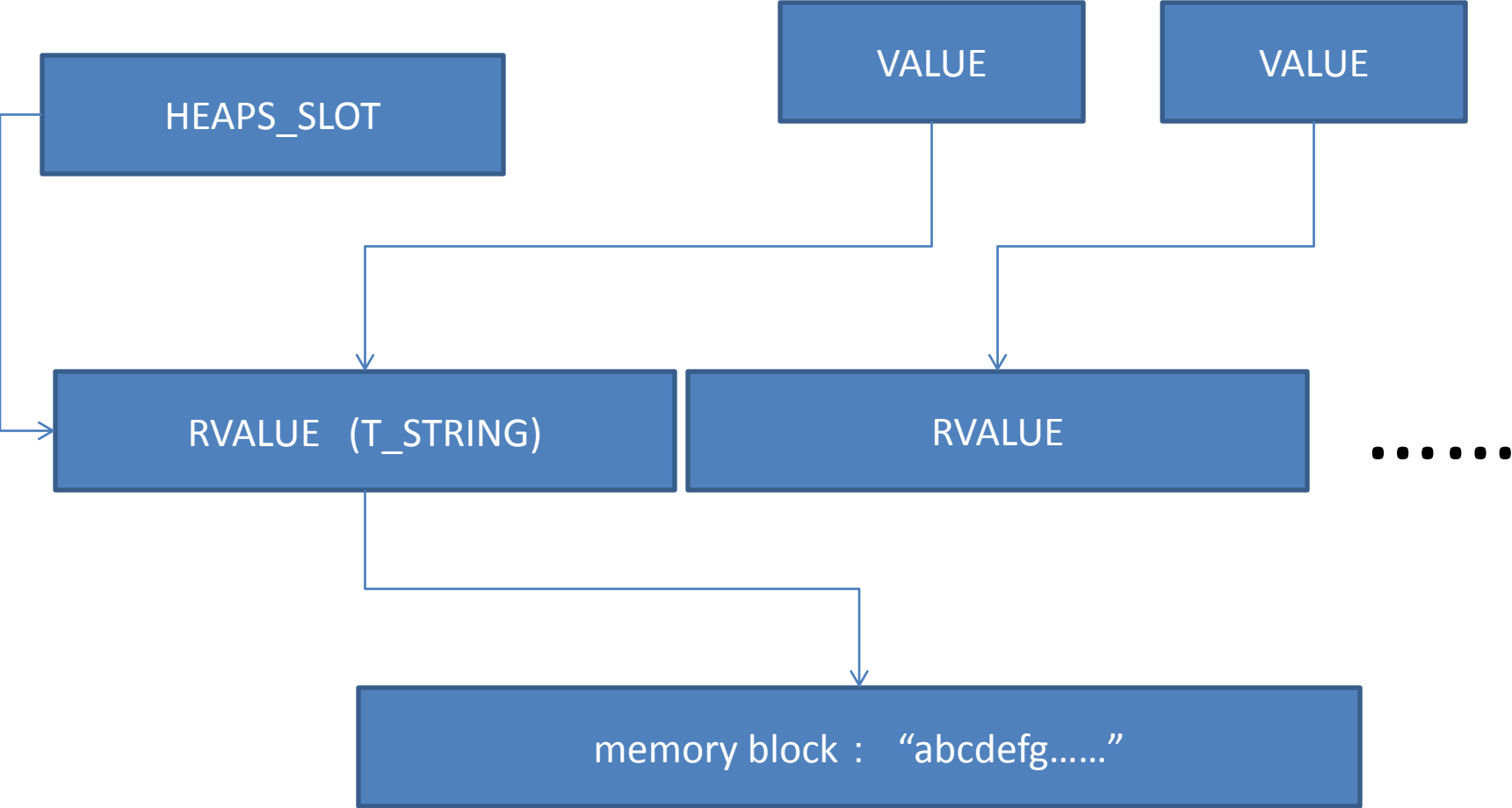


デモ

- inspired by prof. Nakanishi's Apple Lisp
from the article on the old magazin (bit)
- HeapShow
 - <http://svn.arton.no-ip.info/raa/trunk/HeapShow>



RVALUE



苦労したところ＝考慮点

- gc.cのstatic変数やstatic関数
 - externにするのは簡単
 - でも、それでOK?
 - どうせ、依存性あるし.....
- なぜ拡張ライブラリにするのかを考える
 - Do you need yet another Ruby ?
 - or you only need some other (maybe temporary) feature ?

代替案

- dl
 - native API for any external libraries
- COM component
 - win32ole
 - Windows only
- Java class
 - Rjb etc
 - JRuby

dl sample

```
require 'dl/win32'

GetShortPathName = Win32API.new('Kernel32.dll',
                                'GetShortPathNameA', 'SSI', 'I')

olen = 200
begin
  buff = ' ' * olen
  len = GetShortPathName.call(realpath.to_s, buff,
                             buff.size)
  if olen < len
    olen = len
  end
end while olen == len
buff.rstrip.chomp("\0")

// note: it seems bit buggy, but it's correct.
```

tips

- 拡張ライブラリをrequireするRubyスクリプトを同時配布する
 - 前処理
 - 依存性の吸収(実際にrequireするライブラリの分岐など)
 - 後処理(自動実行の制御など)
- バージョン番号を埋め込む

まとめ

C programming is fun !

Q&A