継続ベース アプリケーションサーバフレームワーク

フレームワーク対決 Kahua
柴田 知久

Kahua の特徴

- Scheme 言語処理系 Gauche で書かれている
- S 式が基本データ形式である
- コンテンツはオブジェクトデータベースに保存される
- 対話的に開発できる
- 継続ベースでページ遷移を記述できる

1 Gauche

川合史朗氏によるオープンソースの Scheme 言語処理系でライブラリーが豊富。

モジュールシステム、オブジェクトシステム、マルチバイト文字列、XML パージングなど、一通りの機能が揃っている。

2 S 式

いわゆる Lisp の基本データ形式で、大量の括弧を使用する。

S 式の定義は次の 2 点だけ。

- **アトムは S 式** 1, #t, “string”, symbol …
- **S 式のドット対は S 式** (print 1) (+ 1 2)…

Kahua ではロジックはもちろん、コンテンツ記述言語としても S 式を使う。

■ HTML

```
<html>
  <head>
    <title>Title here </title>
  </head>
  <body bgcolor="gray">
    <h1>Hello</h1>
  </body>
</html>
```

■ S 式

```
(html:
 (head:
  (title: "Title here"))
 (body: (@: (bgcolor "gray"))
        (h1: "Hello")))
```

デザインのテンプレート化も簡単

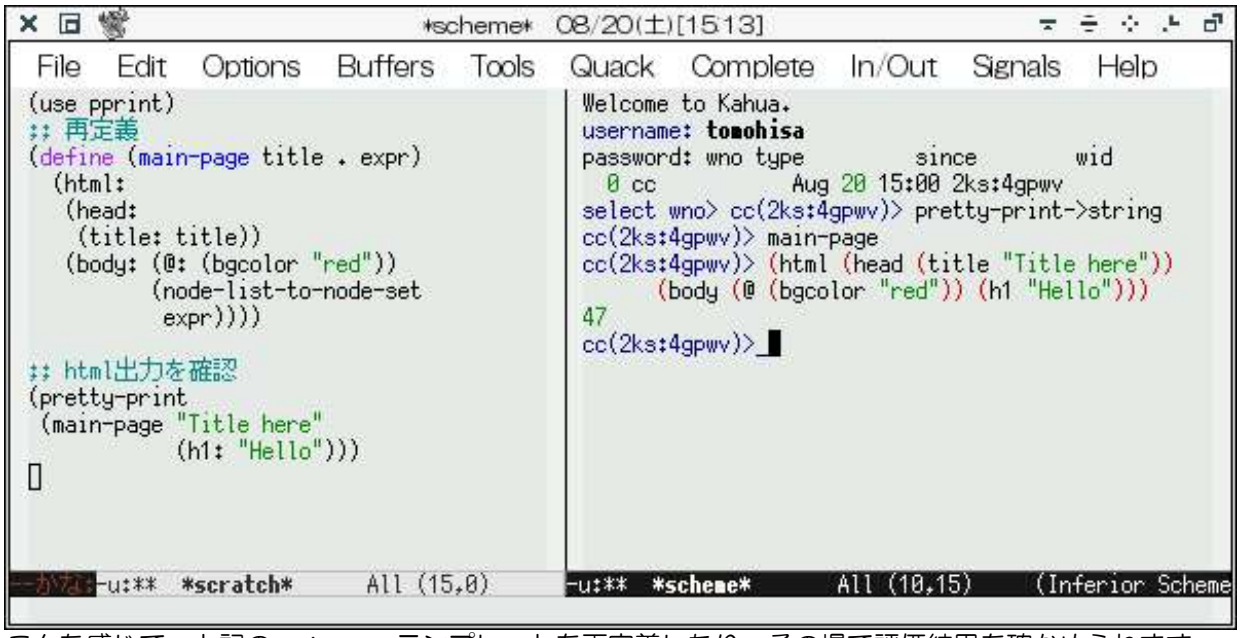
```
;; main-page テンプレートを定義
(define (main-page title . expr)
  (html:
   (head:
    (title: title))
   (body: (@: (bgcolor "gray"))
          (node-list-to-node-set
           expr))))
;; 使用例
(main-page "Title here"
           (h1: "Hello"))
```

こんな感じで、データと手続きを区別することなく統合できます。MVCで書くかは書き手しだい。

でも、デザイナーさんにS式を書いてもらえるかな…

3 対話的な開発

稼働中のKahuaサーバプロセスへ接続して任意のS式を評価できます。



こんな感じで、上記のmain-pageテンプレートを再定義したり、その場で評価結果を確かめられます。

右側がサーバプロセスへ接続したプロンプト。左側に書いたS式を、接続したサーバで評価しながら開発できます。

4 継続ベース

継続とは、計算処理において「次にする事」です。

Webアプリケーションでは、リンクのクリックやフォームをサブミットした時に「する事」を指定しなければならない。Kahuaにはリンクやフォームに継続を対応付ける手続が用意されていて、「次にする事」をページをまたいで指定できる。

```
(define-entry (inc)
  (define (count n)
    (main-page
     ”カウンタ”
     (span: n)
     (a/cont:
      (@@: (cont (lambda () (count (+ n 1))))
      ”増やす”)))
  (iter 0))
```

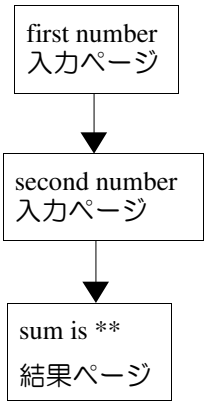
これは「増やす」リンクを押すごとにカウンタを増やす簡単なサンプルです。

継続として(lambda () (iter (+ n 1))を渡しているが、この継続にはその時点の環境も含まれている。

このようにページ遷移を継続渡し形式(CPS)で書けば、ページ間の状態受渡しを気にしなくてよい。

さらに部分継続を利用すれば、暗黙の継続を捕捉できる。
右のように入力ごとに部分継続を捕捉するように書けば

```
(define-entry (sum2)
  (main-page
   ”sum is”
   (p: (+ (read-input ”first number”)
          (read-input ”second number”)))))
```



とページ遷移するアプリケーションになる。