ページ遷移を考慮したWebアプリケーション記述言語の設計と実装

三島　航†　　小宮　常康†　　佐藤　喬†　　多田　好克†
電気通信大学　大学院情報システム学研究科‡

1　概要

Webアプリケーションでは，Webブラウザのバックボタン等のユーザ側の操作によって自由にページを遷移させることができる．そのようなページ遷移は，開発者が想定していない可能性があり，予期しない出力やエラーが表示されてしまう恐れがある．これを防ぐために，開発者は起こりうるページ遷移を全て把握する必要がある．しかし，インタラクションの多いWebアプリケーションにおいて，これは困難である．そこで本研究では，ソースプログラムからページ遷移を容易に把握，解析可能なプログラミング言語の設計と実装を行う．

2　背景

Webアプリケーションは通常のアプリケーションとは異なり，Webページを介して，ユーザとの入出力を行う．Webページはブラウザの持つバックボタン，更新ボタン，URL直接入力等を用いることにより，ユーザが自由に遷移させることが可能である．つまり，ユーザはページを遷移させることで，Webアプリケーションの制御フローを自由に変えることができる．図1は，ユーザ登録アプリケーションにおける制御フローとページ遷移を表したものだが，ユーザは，点線で表すブラウザによるページ遷移を行うことで，実線で表すプログラムの制御フローに自由に入り込むことができる．そのため，開発者の想定していないページ遷移により，Webアプリケーションは開発者の意図しない動作をする恐れがある．意図しない動作の具体例として，ショッピングサイトでの2重注文や誤った注文等があり，問題視されている[1, 2].

想定していないページ遷移を発見するために，開発者は，図1のようなアプリケーション実行中に起こりうる全てのページ遷移を把握する必要がある．しかし，インタラクションの多いWebアプリケーションにおいて，これは困難である．

そこで本研究では，開発者が，ソースプログラム中にページ遷移をある程度書き表すことで，ページ遷移を把握しつつ開発が行え，ソースプログラムからアプリケーションで起こりうる全てのページ遷移を静的に解析できるプログラミング言語の設計と実装を行う．

3　ページ遷移を容易に解析するための必要事項

ソースプログラムからページ遷移を把握しつつ，開発が行え，ソースプログラムからページ遷移を解析できる言語を設計，実装するため，ソースプログラムからページ遷移をどのように把握するのかを考える．ページ遷移を把握するには，プログラムの制御フローを調べる必要がある．しかし，CGIによるWebアプリケーションは，Webページを出力する際に処理を終了し，再開する際に処理を最初から開始するため，

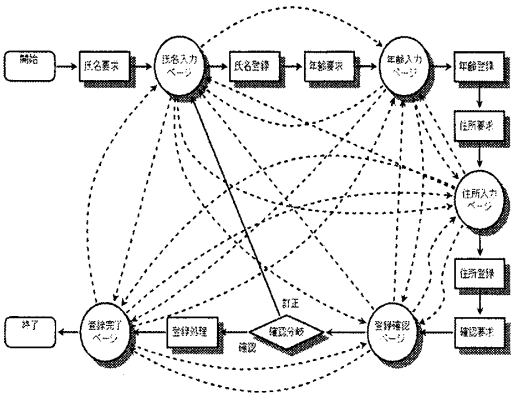

図1: ユーザ登録アプリケーションにおける制御フローとページ遷移 (実線：制御フロー，点線：ブラウザによるページ遷移)

Webアプリケーションの制御構造は複雑になり，ページ遷移を把握することは煩わしい．

制御構造を簡潔に記述するために，継続を用いたWebアプリケーション開発[1, 2]がある．継続とはある時点での残りの処理である．継続を用いることで，処理を最初からではなく途中から再開できるので，アプリケーションのダイレクトスタイルでの記述が可能になる．このため，プログラムの制御構造は簡潔なものとなる．しかし，ページとページの境目がプログラムの記述から把握しにくい．また，制御構造において動的に決定される部分がある場合，ページ遷移の解析は不可能になってしまう．

そこで本研究の言語では，継続を用いつつ，プログラムの記述からページ遷移の把握を容易にするために，個々のページを記述するためにページ定義構文を言語に導入する．そして，ページの実行順序である制御構造を記述するために，ページ遷移構文を言語に導入する．これらの構文のみを用いてプログラムを記述すれば，プログラム実行前にプログラムの記述からページ遷移が静的に決定できるように設計を行う．こうすることで，ページ遷移の把握，解析が容易に行える．

ページ遷移の把握，解析後，想定していないページ遷移を制限し，制限したものを解析結果として表示できるようにするため，ページ定義構文に制限の内容を容易に把握，解析できるパラメータをもたせ，このパラメータをチェックすることでページ遷移を制限する機能を言語に導入した．

4　設計

4.1　ページ定義構文

ページ定義構文は，個々のページを記述するために使用する．ページ遷移を制限するかどうかをチェックするためのパラメータをもたせる．下記がページ定義構文の使用例であり，各種パラメータの説明を表1に示す．

```
(define-page (page-tag (from...)(parameter...)(input...))
ページを出力するコード)
```

Design and implementation of a web application description language with page transition constructs
†Wataru Mishima, Tsuneyasu Komiya, Takashi Satou and Yoshikatu Tada
‡Graduate School of Information Systems, The University of Electro-Communications

表 1: ページの持つ各種パラメータ

パラメータ	用途
page-tag	ページの識別を宣言
(parameter (*type variable*)...)	ページで使用する型と変数を宣言
(input *variable*...)	ユーザ入力を宣言
(from *page-tag*...)	ページ遷移を宣言

表 2: ページ遷移構文

構文名	用途
page-begin	複数のページ遷移構文を左から実行
page-if	単一分岐を作成
page-while	ループを作成
page-cond	多重分岐を作成
page	ページ定義構文で定義したページ

page-tag は，ページ名を宣言する．parameter は，ページ内で使用する変数名と型を宣言する．型には，int，float，string が指定できる．input は，そのページで受け取るユーザ入力を宣言する．*variable* への入力がないならばエラーとなる．from は，ページ遷移のチェックに使用する．*page-tag* には，遷移を許可するページのページタグ，あるいは自分以外の他のページ全てからの遷移を許可する any が入る．

4.2 制限チェック機能

ページ定義構文に設けたパラメータをチェックし，アプリケーションの誤った動作やエラーにつながるページ遷移を制限をするために，以下の 2 つのチェック機能を言語に導入した．

バックボタンや更新ボタンを用いたページ遷移における予期しないエラーをなくすために，特定のページ遷移以外を実行時にエラーと見なす機能を言語に導入した．ページ定義構文が持つパラメータである (from *page-tag*...) をチェックし，許可されていないページからの遷移の場合，エラーページを表示する．

不正なデータが送信されることによる予期しないエラーをなくすために，ユーザ入力を必要とするページで，ユーザから入力を受け取っているか，また，受け取った入力が処理に必要なデータ型であるかをチェックする機能を言語に導入した．ページ定義構文が持つパラメータである (input *variable*) と (parameter (*type variable*)) をチェックし，ページから送信されてくるデータがない，または型が違う場合，エラーページを表示する．

4.3 ページ遷移構文

アプリケーションの制御構造を記述するために使用する．**表 2** に本研究の言語に導入する構文を示す．下記はページ遷移構文の定義を BNF 記法により記述したものである．

```
<ページ式>::=(page-begin <ページ式>+)
        |(page-if <条件> <ページ式> <ページ式>)
        |(page-while <条件> <ページ式>)
        |(page-cond (<条件> <ページ式>)...(else <ページ式>))
        |(page <page-tag>)
<条件>::=Scheme の式
```

5 ページ遷移解析ツール

インタラクションの多い Web アプリケーションにおいて，開発者のページ遷移把握をサポートするために，ページ遷移構文で記述したアプリケーションの制御構造からページ遷移を解析するツールを導入する．以下の 2 種類の解析を行う．

- アプリケーションの記述によるページ遷移解析
- ブラウザによるページ遷移解析

図 1 の実線部分がアプリケーションの記述によるページ遷移であり，点線部分がバックボタンや更新ボタン等を利用したブラウザによるページ遷移である．この解析結果を用いて，ツールが自動でページ遷移に制限を設ける，あるいは開発者が自分でページ遷移に制限を設けていくことで，開発者の意図通り動作する Web アプリケーションを作成する．

6 本研究の言語によるアプリケーション記述

本研究の言語を用いて，2 つの数字をユーザに入力してもらい，その合計を出力するアプリケーションを作成した．そのコードを下記に示す．

```
(define-page (p0 (from any)(input num))...)
(define-page (p1 (from p0)(parameter (int num))(input num2))...)
(define-page (p2 (from p1)(parameter (int num2)))...)

(page-begin (page p0) (page p1) (page p2))
```

アプリケーションの記述から p0，p1，p2 とページが遷移していくのが分かる．ページ p1 では，3 つのチェックが行われる．(from p0) により，ページ p0 からページ遷移してきたかをチェックされる．(parameter (int num)) により，ページ p0 から渡された変数名 num というデータが整数型であるかをチェックされる．(input num) により，ページ p0 で変数名 num というデータが入力されたかチェックされる．

7 おわりに

本研究では，想定外のページ遷移を引き起こすコードやユーザ操作を容易に制限するプログラミング言語の設計と実装を行った．

今後の課題として，本研究の言語ではページ遷移を制限する際に，ユーザに表示されるエラーページを処理系で用意しているが，これを開発者が自由にエラーページ内容を決定でき，エラーページの遷移もページ遷移解析の結果として表示可能にする予定である。

謝辞

本研究の一部は，科学研究費補助金 (20500028) の補助を受けている．

参考文献

[1] Christian Queinnec, Inverting back the inversion of control or, continuations versus page-centric programming, *ACM SIGPLAN Notices*, v.38 n.2 pp.57–64, February 2003.

[2] S. Krishnamurthi, P. W. Hopkins, J. Mccarthy, P. T. Graunke, G. Pettyjohn, and M. Felleisen. Implementation and use of the PLT Scheme Web server. *HOSC, 20(4)*:pp.431–460, 2007.