プログラミングの共有@Wiki

山之上 卓，小田 謙太郎，下園 幸一

Wiki の上でプログラミングを共有できるシステムを試作したことについて述べる．このシステムは PukiWiki と PukiWiki からプログラミング環境を起動するためのプラグインと Java のクラスなどで構成される．デフォルト権限の Java Applet は、ローカルにデータを保存することを許可されていない．本システムを作成するにあたり、遠隔でのデータ入出力を容易かつ統一的に行うため Applet のデータ入出力 API を利用した。この API の上で、保存されるデータをオープンにアクセス可能な Wiki ページとして保存し，Applet の起動のための Wiki タグを導入することより、Java で記述されたプログラミング環境と Wiki システムとの自然な統合を行った．この API を利用することにより，テキストデータの読み書きをする Java プログラムであれば，わずかに書き換えてコンパイルし，クラスファイルを特定のディレクトリに配置するだけで，その Java プログラムを PukiWiki から利用することが可能になる．このとき，PukiWiki のプラグインなどを新たに追加したり，PukiWiki の PHP プログラムを書き換えたりする必要はない．

Sharing Programming @ Wiki

Takash Yamanoue, Kentaro Oda, Koichi Shimozono

Experimental development of systems which can share programming among people in a Wiki, is shown. A programming environment of them can be running on the wiki and it can save its data on the wiki. These environments consist of PukiWiki which is a popular wiki in Japan, the plug-in which starts up Java programs and classes of Java. A Java Applet with default access privilege cannot save its data at the local host. We have constructed an API of Applets for easy and unified data input and output at a remote host. We also combined the API and the wiki system by introducing a wiki tag for starting up Java Applets. It is easy to introduce new kinds of applications with PukiWiki-Java connector. We have used the API for development of these programming environments.

1. はじめに

共同作業は楽しい．業務で行う共同作業の場合，苦痛になる場合もあるが，本来は楽しいものである．趣味の世界でも共同作業は良く行われており，日本でも古くから連歌など，コミュニティに参加している者同士で，共同で作品を作っていく文化がある．

コンピュータネットワークの普及は共同作業を行うときに障害となっていた距離と時間の壁をある程度克服することにも役立っている．インターネットが普及する以前に，パソコン通信が流行していたことがあった．パソコン通信内のコミュニティにより LHA などの有用なソフトウェアの開発が行われた [10]．筆者が参加していたコミュニティ(Nifty Serve, FPL)では，顔を一度も合わせたことがないもの同士で，プログラミング言語 Oscal とその処理系を開発した [1)2)]．連歌の手法を絵画に応用した連画も行われた [3)]．オープンソースソフトウェアの開発も，コンピュータネットワーク上のコミュニティによる共同開発として行われる場合が多い．

WWW は多くの人々がコンピュータネットワークの利用する時の敷居を大きく下げた．それでも当初，WWW で情報発信を行う場合，Web サーバ上に HTML で記述されたファイルを配置する必要があり，この障壁により，多く

の人々は情報発信をためらっていた．しかしながら，その後，ブログや Wiki や SNS などの CMS(コンテンツマネージメントシステム)が普及した．これは情報の送り手と受け手の垣根を低くする技術の一つであり，多くのネットワーク利用者が情報の受信と同じくらい気楽に情報の発信を行うようになった．

Wiki は web ページの上でページの編集ができ，共同作業や情報共有の手段として有効であり，Wikipedia[30] のような影響力の大きなサービスでも利用されている．情報処理学会でも Wiki を利用して活動している研究会がある [24)25)]．日本で良く使われている Wiki クローンの一つとして PukiWiki[31]がある．実際の共同作業では，文書と同様に，絵も共有したいと思うことがしばしば生ずる．筆者が所属するコミュニティのための PukiWiki を使ったサイト(Kumikomi[32])で，絵を使った説明ページを作成する必要が生じた．PukiWiki では，ビットマップの画像を作成するペイントプログラムのプラグイン paint がすでに利用できるが，線や多角形などの画素を配置して絵を作成するドロープログラムのプラグインは見当たらなかった．そこで，SOLAR-CATS[6)7)14)]に備わっているドロープログラムを PukiWiki で利用できるようにするプラグインと，そのプラグインとお絵かきプログラムを連携するためのアプレットを作成し，NetDraw と名付けた [11)15)]．

NetDraw を参考にして，お絵かきだけでなく,作曲して演奏するための簡単な環境や，様々な Java プログラムの起動とそのデータ保存を可能とする Wiki システム，PukiWiki-Java Connector[17)20)]を試作した．このシステムは PukiWiki と PukiWiki から Java プログラムを起動するためのプラグインと Java のクラスなどで構成される．データを文字列として読み書きできる Java プログラムであれば，わずかに書き換えてコンパイルし，クラスファイルを特定のディレクトリに配置するだけで，その Java プログラムを PukiWiki から利用することが可能になる．このとき，PukiWiki のプラグインなどを新たに追加したり，PukiWiki の PHP プログラムを書き換えたりする必要はない．PukiWiki-Java Connector を使うことにより様々な Java プログラムとそのデータが，ネットワーク上で共有可能となる．

Pukiwiki-Java Connector を使って，Java で記述された複数のプログラミング環境を PukiWiki から利用できるようにした．これらの環境で作成されたプログラムは PukiWiki のページに保存することができる．保存されたプログラムを実行し，修正し，再度保存することができる．これらを使って様々なプログラミング言語のプログラミングがネットワーク上で共有可能となる．

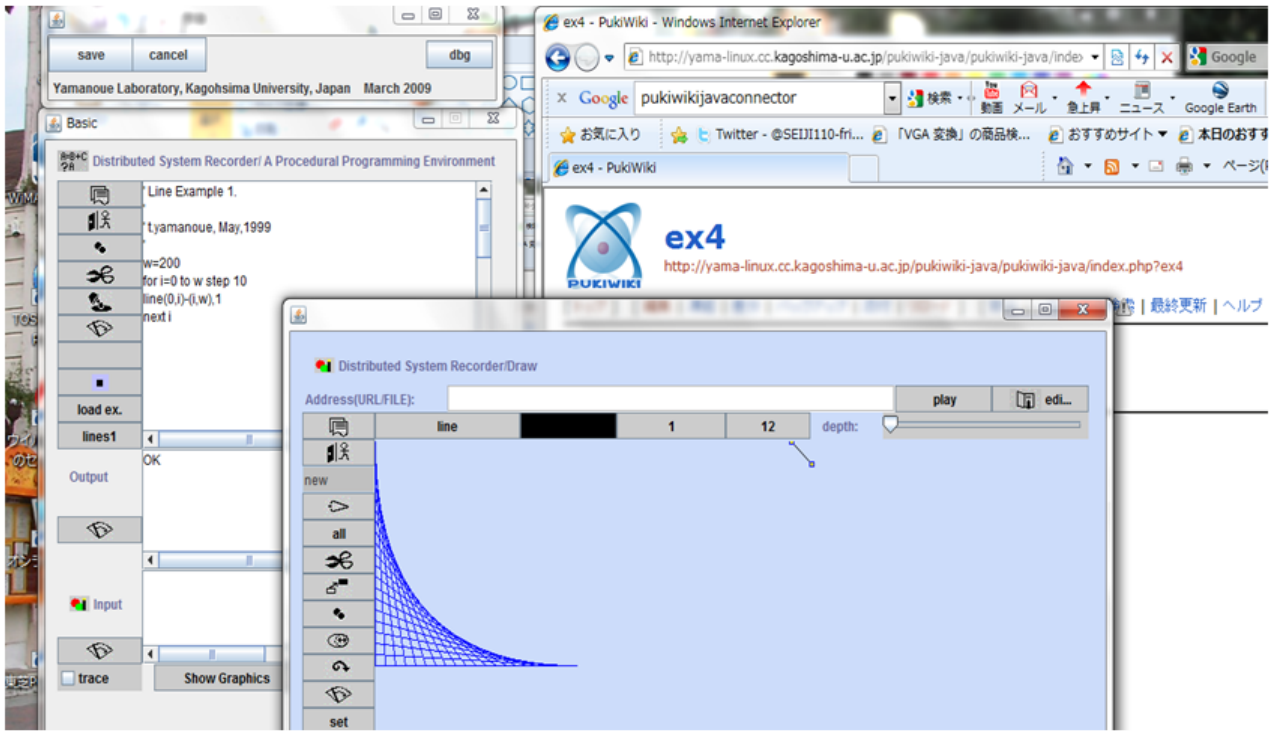

図 1. PukiWiki から起動した Basic に類似した言語のプログラミング環境

図 2. プログラミング環境を起動し，作成されたプログラムを保存するための Wiki ページの作成

図 3. 起動されたプログラミング環境

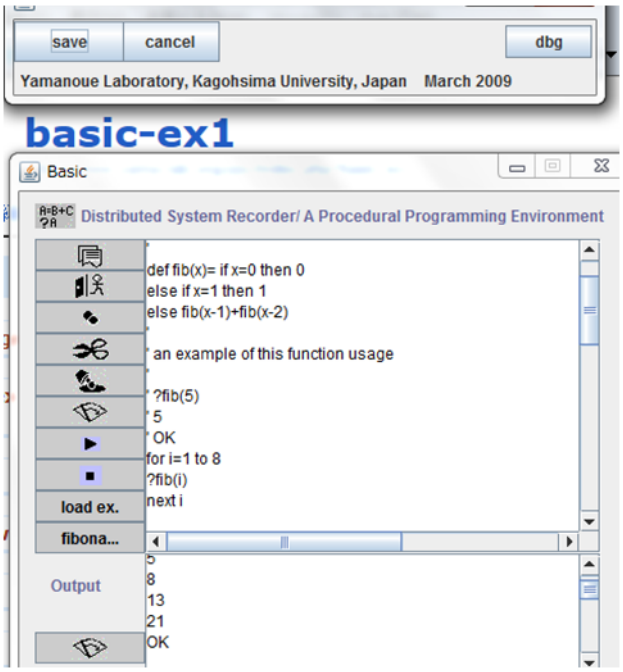

図 4. プログラミングと実行と保存

2. *Basic っぽい言語のプログラミングの共有*

PukiWiki 上で Basic に類似した言語のプログラミングを共有できるシステム(図 1)を例に，このシステムを使ったプログラミングの共有を説明する．

2. 1. **プログラミング環境を起動したり，その環境で作成されたプログラムを保存したりするための PukiWiki ページを以下のようにして作成する** (図 2)

① PukiWiki-Java Connector を導入した PukiWiki で、「新規」ボタンか、「編集」ボタンをクリックし、編集用テキストエディタのページを開く

② 上のページでどこかの行の左端から PukiWiki のブロック型プラグインタグ

#jcon(basic)

を記述する

③ 「ページの更新」をクリックする

2. 1. 「ページの更新」をクリックすると、プログラミング環境が起動する(図 3). 図 3 において，左上に表示されている横に細長いウィンドウは，テキストエディタで作成・編集した内容を保存するときに使う「SaveButtonFrame」である．その下の，タイトルバーに「Basic」と表示されている Java Frame のウィンドウがプログラミング環境である．PukiWiki の背後に表示されているウィンドウは，Web ページで Java Applet を実行するときに表示される Java コンソールである．

2. 2. このプログラミング環境でプログラミングを行う(図 4).

2. 3. SaveButtonFrame の「SAVE」ボタンをクリックすると、作成されたプログラムが、開いている Wiki のページに保存される。 再度、同じページを開くと保存したプログラムが入った GUI が開く

以上のようにして，PukiWiki のページを開いてプログラミング環境を起動し，このとき，保存されたプログラムがある場合はそれが自動的にプログラミング環境に反映され，プログラムの作成，実行，デバッグを行い，SAVE ボタンをクリックして保存することを繰り返す．これを

不特定多数の利用者やグループ内で行うことにより，プログラミングを共有することができる．

3. コピペ

大学生がレポートを作成するときは禁止されている場合が多いが，コピー・アンド・ペースト(コピペ)は，実績のある既存の部品を使うことにより，信頼性のある，良いプログラムを早く作成するときの基本である．

コピペは情報共有・情報拡散の有力な手段の一つでもある．

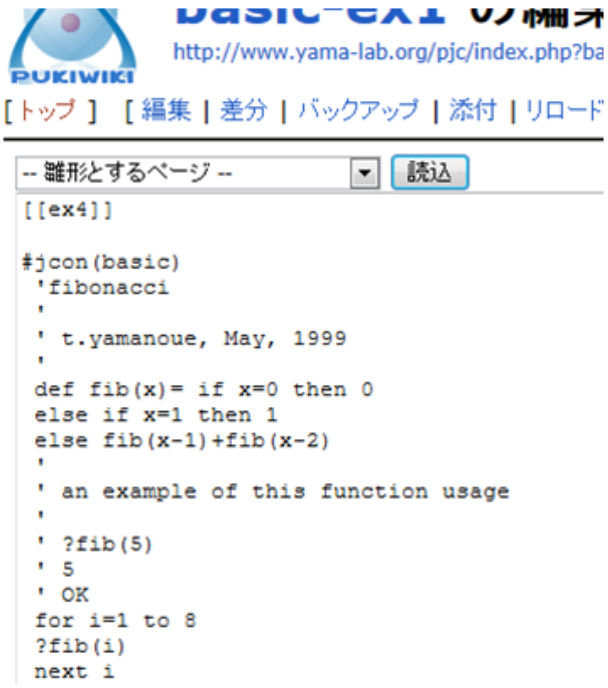

図 5. PukiWiki 編集画面に表示されたタグとプログラム

図 6. 手元の OS でプログラムを選択し，コピー

本プログラミング環境は PukiWiki の機能をそのまま使ってコピペを行うことができる．

図 4 で作成されたプログラムを，新しいページを作成してそこにコピーする場合，以下のような手順を行う．

3. 1. プログラミング環境上のプログラムを保存した後，PukiWiki のメニューの「編集」ボタンをクリックする．

3. 2. ブロック型プラグインタグ #jcon(basic) と，その下に表示されているプログラムをマウスでドラッグして選択し，手元の OS のメニューを表示・選択するなどしてコピーする

3. 3. PukiWiki-Java Connector を導入した PukiWiki で、「新規」ボタンか、「編集」ボタンをクリックし、編集用テキストエディタのページを開く (図 5)

3. 4. 開いた PukiWiki のテキストエディタのページで，手元の OS メニューを表示・選択するなどして 2 でコピーした内容をペーストする(図 6)

3. 5. PukiWiki のティストエディタの「ページの更新」ボタンをクリックする．これによりペーストした内容が新しいページに保存され，一緒にコピーされたブロックプラグインタグにより，プログラミング環境が起動され，起動されたプログラミング環境にコピーしたプログラムが表示される．

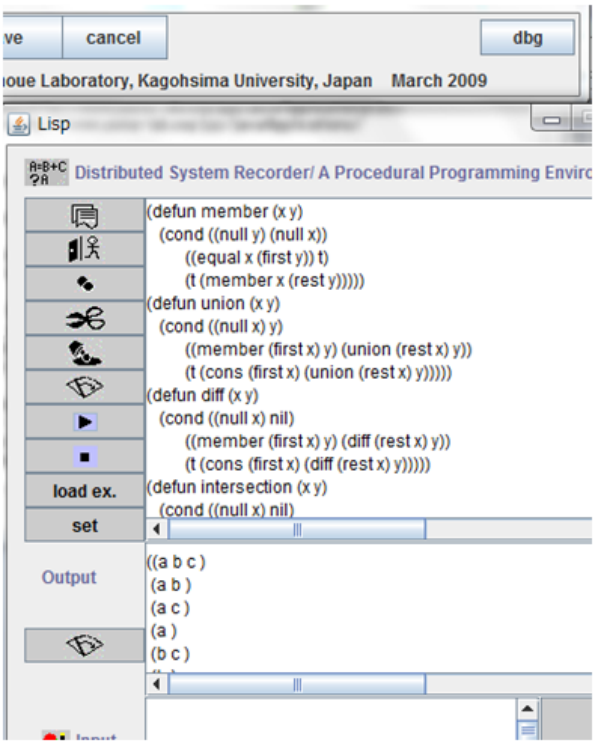

図 7. 簡単な Lisp のプログラミング環境

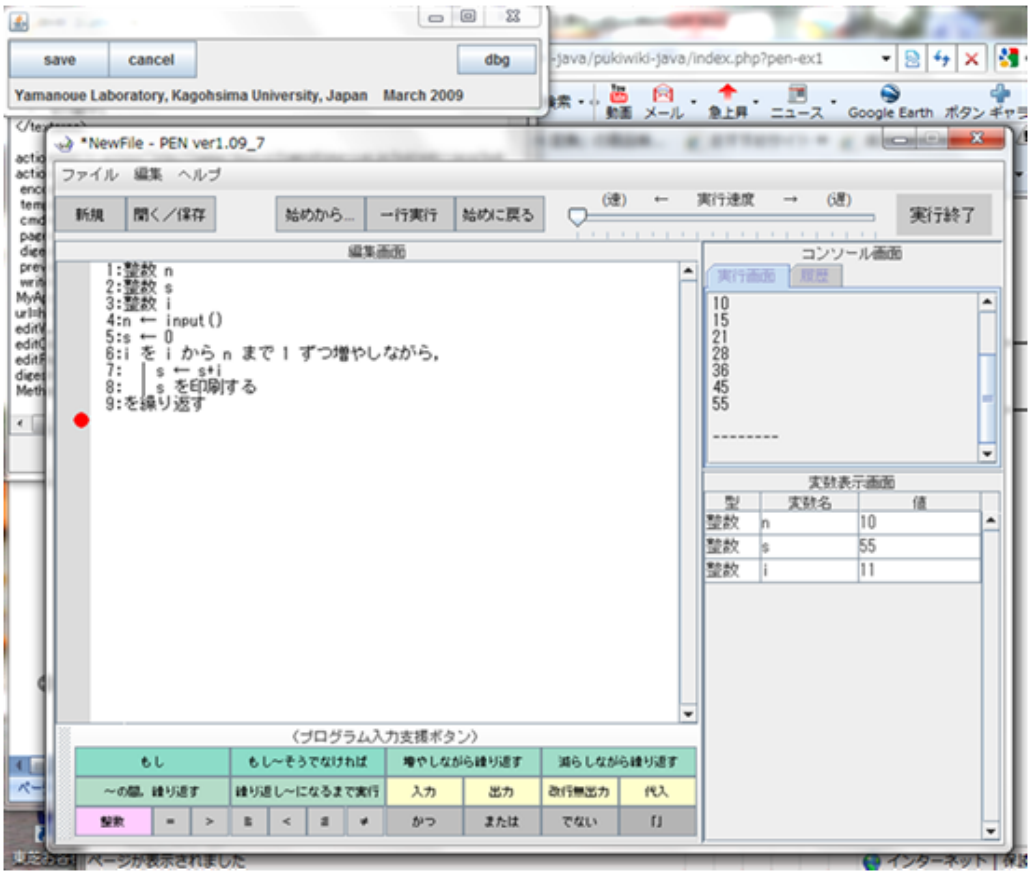

図 8. PukiWiki に組み込まれた教育用プログラミング環境 PEN

上の例では，プログラミング環境起動タグも含めて，プログラム全体をコピーしたが，プログラムの一部のみコピペすることも可能である．

4. その他のプログラミング環境

Basic に類似した言語のプログラミング環境の他に，簡単な Lisp のプログラミング環境(図 7)や，日本語による教育用プログラミング環境である PEN (図 8)も PukiWiki 上で利用できるようにした．Basic に類似した言語のプログラミング環境は，内部ではプログラムを Lisp の内部表現に変換し，それを Lisp インタープリタで実行している．簡単な Lisp のプログラミング環境は，Basic に類似した言語のプログラミング環境において，外部表現をS式に置き換えているだけである．

PEN は大阪学院大学・大阪市立大学で開発された xDNCL のプログラミング環境である．xDNCL は，大学入試センター情報関係基礎の問題で使われるプログラミング言語と東京農工大学の入試用手順記述言語 TUATLE に準拠した言語である．PEN も Java で開発されており，他のプログラミング環境と同様にして PukiWiki に組み込むことができた．

5. PukiWiki-Java Connector

PukiWiki-Java Connector を使って，プログラミング環境の他，例題として，簡単なテキストエディタ，簡単な音楽データ編集とその演奏環境，お絵かきプログラム(NetDraw)，ボイス・レコーダを PukiWiki に埋め込むことができた．Pukiwiki の編集ページで左端からブロック型プラグイン呼び出し

#jcon(<Java アプリケーション名>)

を記述することにより，そのページを表示したときに <Java アプリケーション名>で指定された Java プログラムが起動される．そのプログラムが読み込んだり，保存したりするデータは，このページの，プラグイン呼び出しの次の行から記述される．

(ア) インストール

PukiwikiJavaConnector を簡単に利用できるようにするため，Pukiwiki に PukiwikiJavaConnector を組み込んだファイル(pukiwiki-java.tar.gz)を用意している．このファイルをダウンロードし，通常の Pukiwiki と同様にインストールすれば，PukiWikiJavaConnector が組み込まれた PukiWiki を利用することができる．

既存の PukiWiki に PukiWiki-Java Connector を加えるためには，PukiWiki-Java Connector のファイル(javaApplications.jar)をダウンロードして解凍し，解凍された javaApplications ディレクトリとその中身を Pukiwiki のディレクトリの直下にコピーする．使用する PukiWiki のトップディレクトリを <pukiwiki> で表すと，<pukiwiki>/javaApplications となる．次に，<pukiwiki>/plugin ディレクトリに，ブロック型プラグイン jcon.inc.php を追加する．jcon.inc.php の内容を図 9 に示す.

```
<?php
// PukiWiki - Yet another WikiWikiWeb clone
//
// jcon.inc.php
//              t.yamanoue, 2010
// …

function plugin_jcon_convert()
{
        if (PKWK_READONLY) return ''; // Show
nothing
        $args = func_get_args();   //引数
        if (count($args) >= 1) { $aw =
array_shift($args);} else { $aw = 'draw';}
    $java_application_name =
htmlspecialchars($aw,ENT_QUOTES);
        $ret = ''; //戻り値
    $charset=CONTENT_CHARSET;
    $uri=get_script_uri() ;

$jcode="application.".$java_application_name.".MyAp
plet.class";
    $plginname="jcon(".$java_application_name.")";
        $ret = <<<EOD
<div>
<applet codebase="./javaApplications/bin"
code="$jcode"

archive="lib/commons-codec-1.3.jar,lib/commons-httpc
lient-3.1.jar,

lib/commons-logging-1.1.1.jar"
        width="100" height="100">
  <param name="action" value="$uri" />
  <param name="param1" value="plugin=$plginname"
/>
  <param name="charset" value="$charset" />
  </div>
EOD;
        return $ret;
}
?>
```

図 9.　jcon.inc.php

(イ) 新しい Java プログラムの追加

データの保存・読み込みがテキスト形式で行うことができれば，そのような既存の Java プログラムを少し書き換えてコンパイルし，クラスファイルを特定のディレクトリに格納することにより，PukiWiki のページからそのプログラムを起動することが可能になる．このとき，PukiWiki の php プログラムを変更する必要はない．一種の Factory Method Pattern[4] を使うことにより，新たなプログラムを追加するときの書き換えを少なくしている．図 6 に NewAppli という名前の Java プログラムを追加する場合の，UML のクラス図を示す．新しい Java プログラムを追加する場合以下のように行う．

①　ソースディレクトリの作成

pukiwiki-java ディレクトリの内に，javaApplications というディレクトリがある．この中に，src という，java のソースプログラムを格納したディレクトリがある．新規に application を追加する場合は，src の中の application ディレクトリの中に，新しいディレクトリを作成する．このディレクトリの名前がブロック型プラグイン jcon の引数となる．例として，この名前を newAppli とした場合，この application は，PukiWiki の編集ページの左端で

#jcon(newAppli)

を書くことにより，起動されるようになる．また，このディレクトリは，パッケージ application.newAppli を表すことになる．以下，newAppli の場合で説明する．

②　既存の MyApplet.java のコピーと変更

newAppli ディレクトリに，application.myEditor パッケージ (src の下の application の下の myEditor ディレクトリ)内の，MyApplet.java をコピーする．コピー後は

<pukiwiki>/javaApplications/src/application/newAppli/MyApplet.java

のようになる．MyApplet.java のパッケージ宣言を application.newAppli に変更し，getPukiwikiJavaApplication メソッド内の new MyEditor() を new NewAppli() に書き換える．コンストラクタの引数は，追加する java application に応じて追加する．getPukiwikiJavaApplication メソッドが factory method である．

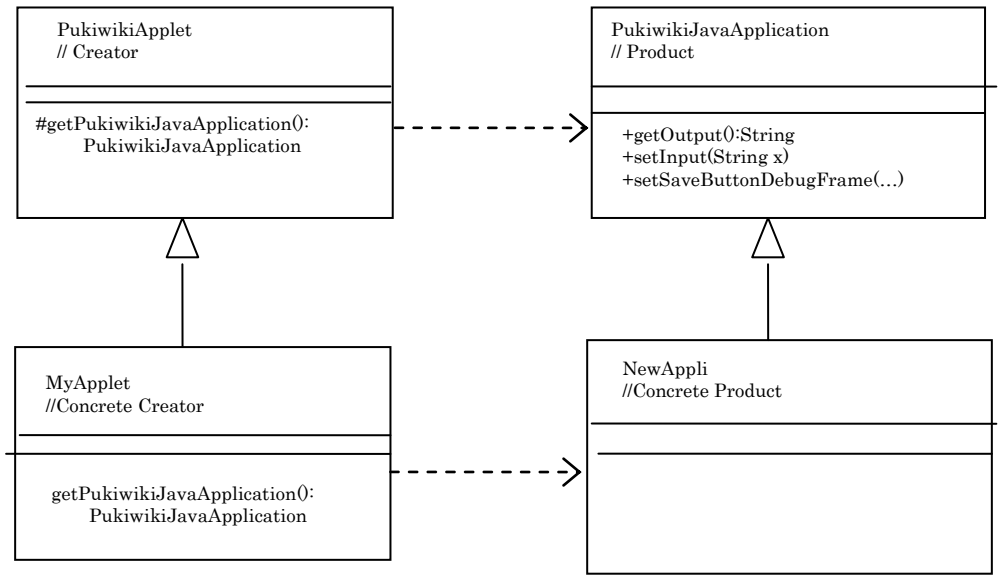

図 6. NewAppli を追加する場合のクラス図

```
package application.newAppli;
public class MyApplet extends PukiwikiApplet{
        public PukiwikiJavaApplication
        getPukiwikiJavaApplication(){
            if(frame==null){
               frame=new MyEditor();
            }
            return this.frame;
        }
}
```

③ 追加するアプリケーションのコピー

追加するアプリケーションを newAppli ディレクトリにコピーする．ここで，このアプリケーションは，JFrame 型のクラスを持っている必要がある．このクラスが PukiWiki から起動される．たとえば，このクラス名を NewAppli とする．このプログラムは，NewAppli.java となる．NewAppli.java をコンパイルするのに必要な他の java ファイルも，このディレクトリなどにコピーする．

```
package application.newAppli;
…
import connector.*;
public class NewAppli extends
JFrame implements PukiwikiJavaApplication
{
   ...
}
```

④ インターフェースの implements

NewAppli.java のパッケージを application.newAppli とし，PukiwikiJavaApplication インターフェースを implements する．

PukiwikiJavaApplication を implements するために必要な，以下の３つの method を NewAppli.java に加える．

```
@Override
public String getOutput() {
        // TODO Auto-generated method stub
}
@Override
public void setInput(String x) {
        // TODO Auto-generated method stub
}
@Override
Public void setSaveButtonDebugFrame
    ( SaveButtonDebugFrame f )  {
        // TODO Auto-generated method stub
}
```

必要に応じて，これらの method の中身を埋める．getOutput() は，PukiWiki のページに保存する NewAppli のデータを取り出す．「SAVE」ボタンがクリックされたとき，この method が呼び出され，NewAppli のデータがこの method を通じて取り出され，Pukiwiki のページに保存される．

setInput(String x) は，Pukiwiki のページから NewAppli が起動された後，そのページに書かれた NewAppli のデータを取り出した後に呼び出される．引数 x が取り出され

たデータであり，これを NewAppli で処理することにより，NewAppli にデータが渡されたことになる．setSaveButtonDebugFrame は NewAppli 側で save を呼び出したいときに利用する．

以下に，記述例を示す．

```
@Override
public String getOutput() {
        return this.myTextArea.getText(); // add
}
@Override
public void setInput(String x) {
        this.myTextArea.setText(x);    // add
}
@Override
public void setSaveButtonDebugFrame
(SaveButtonDebugFrame f) {
}
```

⑤ コンパイルとコピー

NewAppli ディレクトリ内の Java ファイルをコンパイルし

<javaApplications>/bin/application/newAppli

ディレクトリを作成し，コンパイルで生成されたクラスファイル MyApplet.class, NewApppli.class および，他の class ファイルを，bin/application/newAppli にコピーする．

なお，本システムは apache の httpclient[28] を利用しており，<javaApplications>/bin/lib に httpclient に関する jar ファイルが必要になる．

6. 開発期間

NetDraw を開発したとき，PukiWiki-Java Connector はなかった．このとき，NetDraw の図データを PukiWiki のページから抽出するための構文解析部を作成したり，PukiWiki ページにお絵かきのデータを埋め込むための様々な変換プログラム を作成したり，このアプレットを起動するための PHP プログラムを作成したりする必要があった．PHP や PukiWiki のプログラムを学習する時間も必要であった．NetDraw の開発には約 3 週間/人の時間が必要であった [11][12]．

PukiWiki-Java Connector を使った場合，NetDraw を開発するために必要であった様々なプログラムはあらかじめ javaApplications パッケージに格納されており，これらを Factory method パターンを使って簡単に利用できるようになった．PukiWiki 側のプログラムも書き換えたり追加したりする必要はない．NetDraw と同じプログラムを PukiWiki-Java Connector を使って PukiWiki に組み込んだ場合，必要な時間は 3 日/人程度であった [17]．簡単なテキストエディタの場合は 2 時間/人程度，プログラミング言語実行環境や音楽編集・演奏プログラムの場合は 1 日/人程度であった．Java の開発環境として Eclipse を利用 した．

7. 関連研究

共同でプログラミングを行う場合，従来から GIT[16]，Subversion[13]，CVS[5] などのバージョン管理システムが利用されている．しかしながら，これらを使う場合，プログラミング環境は手元のホストで動作しているものを利用する．これに対して本論文は Web 上でプログラミングできることを示している．

最近，Web でプログラミングできる，組込みシステムのプロトタイプ作成環境とデバイスを統合した mbed が利用できるようになった [33]．Web でプログラミングできる，という意味において，本論文の内容と mbed は類似している．しかしながら mbed は専用の Web サーバで動いており，利用者が自ら mbed のプログラミング共有サーバを立ち上げることは困難である．これに対して本論文は，一般的な Wiki の上にプログラミング環境を構築したことを述べている．これによって，利用者は比較的容易に自分でプログラミング環境サイトを立ち上げることができる．日本でも AROE[8] や ASPEN[19] などの，Web 上のプログラミング環境がすでに存在する．これらのプログラミング環境も一般的な Wiki の上で使えるものではない．

PukiWiki 用に開発された既存の Java アプレットプラグインもあるが [26]，このプラグインでは起動されたアプレットで作成されたデータを保存したり，そのデータを多くの人で共有したりする機能はない．

8. おわりに

PukiWiki 上でプログラミングの共有を可能にするプラグインを開発したことについて述べた．このプラグインは，様々な Java プログラムの起動とそのデータ保存を PukiWiki に行うための API と，既存の Java で開発されたプログラミング環境を利用することにより，短時間で開発することができた．Basic 認証にも対応しているので，

利用者を制限することも可能である．今後，本プログラミング環境の改良，利用方法のマニュアルの整備，多くの利用者への広報などを行っていく予定である．

謝辞

本論文の内容を実施するために利用させていただいた，PukiWiki と PEN の関係者の皆さまと，NetDraw や PukiWiki-Java Connector 開発にあたりコメントをいただいた皆さまに感謝します．

質疑応答

Q（NTT 研究所 中山）．Wiki ページの作成にしか利用していないように見える．Wiki のうえにあることのメリットがよくみえない。Wiki のうえにあることの利点の事例は?

A. 絵をたくさん使う実験の授業で使っている[18]．実験資料のミスをその場で修正して即座に反映させている．

Q(NTT 研究所 中山)．ページ間連携などの Wiki ならではのインターフェースがないように思える。

A. 今後の課題として考えたい

Q(東京大学 荒川)． 実行を必ずともしたくないなどの場合にそなえて 1 クッションを。

A. 検討する．

Q(東京大学 荒川)．よけいなコンソールがでるがどうにかならないのか?

A. 仕様なのか出てしまう．出ないやり方があれば教えて欲しい．

Q(東京大学 荒川)．ウィンドウを埋め込むとか，必要な場合だけポップアップだとか，インターフェースの改善を。

A. ありがとうございます．了解．

Q(東京大学 荒川)． 立ち上がりがおそいように見える。もっと早くならないものか?

A. 一部を Jar 化することによって少し速くなった．他にも努力するが，Java はもう古いかもしれない．

Q(東京大学 荒川) そう思う．これからは HTML5 などが良さそう．

Q(横浜国大 倉光)．Wiki でやっているとどうやってグループコミュニケーションをとっているのか? どれだけの開発スパン(書き込み間隔)になるのか？

A. まだあまりグループコミュニケーションができていない．これからである．現在は一人がいろいろな意見を聞いて書き換えるのが現実的と思う．

Q(横浜国大 倉光)．ASPEN は Wiki ではないが，twitter と融合させたい．学生は長い文章を書くのが苦手． 学生の演習では twitter を使いたい．Wiki は学生にとって少し重い．現在、実際に使っているのか?

A プログラミング授業ではこれから。お絵かきでは使っている．

Q(九工大 小出)．Wiki を使うと，ひとりでの作業になりやすく，大きな仕事ができにくい気がする．共同作業化にうまく活用するにはどうすればよいか?

A. 今後の夢ではあるが，中山さんから意見があったように Wiki のページ間の連携を使って，みんなで大きなシステムを作ることができるかもしれない．

Q(九工大 小出)．twitter 版の連携システムは難しいか?

A. できなくはないと思う。

参考文献

1) 山之上卓, 紀太章, 相沢雅陽 "パソコン通信を利用したソフトウエアの共同開発 -プログラミング言語 Oscal の設計と実現-", 情報処理学会第 37 回(昭和 63 年後期)全国大会講演論文集(II), 3L-9,pp.812—813, 1988.
2) "パソコン通信を使い電算機言語を開発" 日本経済新聞,11 月 17 日, 1988.
3) 木原 民雄 , 藤井 孝一 , 中村 理恵子 , 安斎 利洋, “ネットワーク共有空間での人間の動きによる描画と演奏”, 情報処理学会論文誌,Vol. 40, No. 9, pp.3501-3509 , Sep. 1999
4) 結城浩, “Java 言語で学ぶデザインパターン入門”, ソフトバンククリエイティブ, 2001.
5) Jennifer Vesperman 著, 滝沢徹, 牧野祐子訳, “実用 CVS”, オライリー・ジャパン, 2003.
6) 山之上 卓: P2P 技術を利用した分散システム上の実時間操作共有システム, 情報処理学会論文誌, vol.46, No.2, p.392-402, 2005.
7) Yamanoue, T., “Sharing the Same Operation with a Large Number of Users Using P2P”, The 3rd International Conference on Information Technology and Applications (ICITA'05), IEEE CS Press, pp.85-88, July 2005.
8) 長 慎也, 兼宗 進, “Aroe - Ajax で Ajax を作る”, 情報処理学会夏のプログラミングシンポジウム 2006.
9) 西田知博, 原田章, 中村亮太, 宮本友介, 松浦敏雄, “初学者用プログラミング学習環境 PEN の実装と評価”, 情報処理学会論文誌, Vol.48, No.8, pp.2736-2747,Aug.2007.
10) アスキー書籍編集部 編, “蘇る PC-9801 伝説 永久保存版 第 2 弾”, アスキー, 2007.
11) 山之上 卓, “PukiWiki 用ドロープログラムのプラグインの試作“, 情報処理学会研究報告インターネットと運用技術（IOT）,2009-IOT-5(5),1-6 (2009-05-21)
12) 山之上 卓, “Pukiwiki 用ドロープログラムのプラグインの試作と教育への応用,” 情報教育シンポジウム論文集, 情報処理学会シンポジウムシリーズ vol.2009, No.6, (IPSJ SIGCE SSS2009), pp.55-60, Aug. 2009.
13) C. Michael Pilato, Ben Collins-Sussman, Brian W. Fitzpatrick 著,

宮本久仁夫監訳，朝枝雅子，浜本階生訳，“実用 Subversion”，オライリー・ジャパン，2009.
14) Takashi Yamanoue, "A Casual Teaching Tool for Large Size Computer Laboratories and Small Size Seminar Classes" , Proceedings of the 37th annual ACM SIGUCCS conference on User services, pp.211-216, St.Louis, Missouri, US.. 11-14 Oct. 2009.
15) Takashi Yamanoue, "A Draw Plug-in for a Wiki Software", saint, pp.229-232, 10th IEEE/IPSJ International Symposium on Applications and the Internet, 2010.
16) Jon Loeliger 著，吉藤英明 監訳，本間雅洋，渡邉健太郎，浜本階生訳，“実用 Git”，オライリー・ジャパン，2010.
17) 山之上卓，山本史弥，小田謙太郎，下園幸一，“Java プログラムの起動とそのデータ保存が可能な Wiki システムの試作”，情報処理学会研究報告，Vol.2011-CE-109, No.13, 2011.
18) 山之上　卓，大橋勝文，　大野裕史，　鹿嶋雅之，　池田亮，“FPGA 評価ボードを使ったインターフェースの実験演習,” 情報教育シンポジウム論文集，情報処理学会シンポジウムシリーズ vol.2011, (IPSJ SIGCE SSS2011), Aug. 2011.
19) 若森 拓馬，中田 晋平，倉光 君郎，“オンラインセキュリティ実験基盤”，情報処理学会夏のプログラミングシンポジウム 2011..
20) Takashi Yamanoue, Kentaro Oda, Koichi Shimozono, "PukiWiki-Java Connector, a Simple API for Saving Data of Java Programs on a Wiki" , ACM WikiSym '11, Proceedings of the 2011 international symposium on Wikis, p.224, Mountain View, CA, USA, 3-5 oct.,2011.
21) オンライン版ドリトル, http://kanemune.eplang.jp/diary/2009-07-12-1.html
22) PEN のアプレット版, http://kanemune.eplang.jp/diary/2009-07-12-1.html
23) PukiWiki 公式サイト, http://pukiwiki.sourceforge.jp/
24) 情報処理学会インターネットと運用技術研究会, http://iot.ipsj.or.jp/
25) 情報処理学会コンピュータと教育研究会, http://ce.eplang.jp/
26) 九州産業大学藤田研究室, http://w3fj.te.kyusan-u.ac.jp/miwiki/
27) Xampp for windows, http://www.apachefriends.org/jp/xampp-windows.html
28) Http Client Home, http://hc.apache.org/httpclient-3.x/
29) NetDraw, http://yama-linux.cc.kagoshima-u.ac.jp/netdraw
30) Wikipedia, http://www.wikipedia.org/
31) PukiWiki, http://pukiwiki.sourceforge.jp/
32) Kumikomi, http://yama-linux.cc.kagoshima-u.ac.jp/kumikomi/
33) Rapid Prototyping for Microcontrollers, http://mbed.org/