

# 演習問題個人化のためのユーザ添削に基づく理解箇所判定手法

## Detection Method of Understanding Part based on the User Correction for Exercise Personalization

鈴木 尚樹<sup>†</sup> 北山 大輔<sup>†</sup>

<sup>†</sup> 工学院大学情報学部 〒163-8677 東京都新宿区西新宿 1-24-2

E-mail: [tj110072@ns.kogakuin.ac.jp](mailto:tj110072@ns.kogakuin.ac.jp), [takitayama@cc.kogakuin.ac.jp](mailto:takitayama@cc.kogakuin.ac.jp)

**あらまし** 人に教える行為の中で添削という学習方法がある。添削により、教える側も知識を再確認し教えることで実は学習すると考えられる。しかしながら教える行為の学習効果は高いと考えられるが、教えられる側を必要とするため e-learning で実現は少ない。そこで、我々はユーザが添削をする式を仮想生徒が自動生成してくれるシステムを提案する。ユーザは、学習者なので添削を間違える可能性がある。そのため、問題中のどこを理解しどこを理解していないのかを添削を行う中で判断し提示する手法を開発する。

**キーワード** e-learning, 添削, 理解箇所判定, プロダクションルール, 仮想生徒

### 1. はじめに

現在の e-learning はさまざまな学習方法が存在する。ゲーム機、スマートフォン、PC などのソフト(アプリ)で学ぶ学習方法や iTunesU<sup>(注1)</sup>などの動画サイトで講義を見る学習方法、Wikipedia<sup>(注2)</sup>、KIT 数学ナビゲーション<sup>(注3)</sup>などの Web サイトなどで情報を閲覧する学習方法がある。

学習に関して良く知られているものに学習のピラミッドがある。これは、学習方法の違いによって学習の定着率がどのように異なるかを示したものである。このピラミッドによると伝統的な講義、読む、教材の定着率は低く、体験を伴うもの、他人に教えることの定着率は高い。このピラミッドの数値に関する根拠は諸説あるようだが、経験的に受け入れられていると考えられる。e-learning でよく使用される学習方法は、定着率が低く、定着率が高い学習方法はあまり使用されていない。そこで我々は、学習のピラミッドの中で定着率が一番高い学習方法である、人に教える学習方法を e-learning に適用する必要があると考えた。

人に教える学習方法として添削がある。添削をするメリットは、知識を再確認することができ、弱点の発見や知識の再構成を行えることである。一般に、添削をするためには、添削される側が必要となる。そこで、添削を行うために必要な式を、仮想生徒が自動的にユーザの理解度に応じたレベルで誤りを含んで解く仮想生徒システムを提案する。この時ユーザは、間違っただけの添削をすることがある。実際の添削では間違っただけの添削にリアルタイムに気づくことは難しい。本研究ではユーザの理解・不理解箇所を調べる理解箇所判定手法を用いることにより、不理解箇所により、間違っただけの添削してしまった箇所を判断できる。

その部分に関してユーザにフィードバックを与えることで、学習を促進させる。このことにより学習者個人でも、添削のメリットである知識を再確認できる部分を活用可能となる。

本稿の構成を以下に述べる。2 節では、本研究のアプローチとして、概要と関連研究を述べ、e-learning における本研究の位置付けを述べる。3 節では、仮想生徒システムの全体像を説明し、4 節では、数式の変形方式。5 節では、理解箇所判定手法について詳細に説明する。

### 2. 本研究のアプローチ

#### 2.1 研究の概要

添削をするメリットは、知識を再確認することができることである。一般に問題を解くだけでは細かい途中式に気を配ることなく解いてしまい、定着率が低いと考えられる。そこで、本研究では添削を行うシステムを提案する。添削を行うには、生徒と教える人が必要である。生徒は、システム上に構築するものとし仮想生徒と呼ぶ。教える人はユーザである。添削を行うには、仮想生徒の間違った式が必要となる。その式は、ユーザの理解度と仮想生徒の式の状態によって生成される。例の式として、 $3x+3 \leq 2x+12$  を出題する。ユーザの理解度は、“符号を逆にすることを理解していない”とする。なお、本研究では理解度は確率で表し理解しているほど出題時にその理解内容に関する誤りが少なくなる。式の状態は、“左辺に同類ではない項がある”である。この時仮想生徒の状態に当てはまる式変更ルールを適用し、ユーザの理解度により  $3x \leq 2x+3+12$  を生成する。この式は、符号を逆にすることを理解していないユーザに対して添削をさせるために誤った式を生成している。ユーザが添削するとき間違えて添削してしまう場合がある。ユーザの添削した式が正しいか間違っているかを確かめるために、ユーザが添削した仮想生徒の解答の直前の式に適用可能なすべての if-then ルールを用いるとする。生成された複数の式とユーザが添削した式をマッチングさせることにより、ユーザの式が正

(注1) : <http://www.apple.com/jp/education/itunes-u/>

(注2) : <http://ja.wikipedia.org/wiki/メインページ>

(注3) : <http://w3e.kanazawa-it.ac.jp/math/>

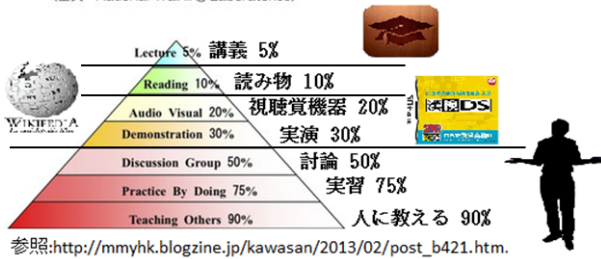


図 1 学習ピラミッドと e-Learning

しいか間違っているかが判り、ユーザの理解度とする。理解度を判ることにより、ユーザの弱点が克服されると考えた。

## 2.2 関連研究

従来の e-learning に関する研究として、以下のような研究があげられる。白井ら [1] の研究では、数式の入力が容易なシステムを用いドリル学習をやりやすいようにしている。中村 [2] の研究では、学習者は KIT 数学ナビゲーションから自分の知りたい知識を探すことができる。知りたい知識の内容には、詳しい説明が載っている。学習者が説明の中で理解できない内容にリンクが張ってあれば、その説明に移ることができ、基本的な内容に立ち返って学習することができる。

新竹ら [3] は、算数の文章題を対象とした作問システムモンサクン 2 に協調的な要素を組み込んだ。協調化の一部として複数人で複数の問題を作り、別のチームと問題を作った時に配分される得点で競うというのがモンサクン 2 に付け加えた要素である。このとき、複数の問題で一番得点を取れるものをユーザが吟味する。従来ではメタ認知が活性化するかどうかは学習者にゆだねられていたが、このとき、複数の問題で一番得点を取れるものを吟味することによって、メタ認知活性化の支援になっている。こうすることによって問題の構造も理解することができる。

岡崎ら [4] の研究は、携帯ゲーム機のタッチペンによる直接操作を学習に活用して、理科の実験・観察・演習問題に適用し、携帯ゲーム機で実験・観察・問題演習を行うシステムを開発した。岡崎ら [4] のシステムは、携帯ゲーム機である NDS 上で動く。NDS はダブルスクリーンになっている。例として、タッチペンを直接操作してプレパラートを作成する。その時スクリーンにプレパラートを作成する指示が表示され、他のスクリーンにプレパラートを作成するための部品が用意されていて、その部品を動かすことができるようになっている。実験を行うには、特別な部品や安全に行える環境が必要であり、そのようなことを考えると個人で実験は容易に行えない。岡崎ら [4] のシステムを使うことにより気軽に何度でも行えるようになる。

本田ら [5] の研究では、問題が表示されておりその問題の答えが正しいと終了する。答えが正しくない場合、学習者理解度の診断の上にヒントを提供し、学習者の理解している部分まで数式を解いてくれる。ただ、問題の答えは、最終的なものを入れるので、解き方が間違っているでも最終的な答えが合えば、正

しいになってしまう。なので、途中の式の内容を理解しているかわからない。我々の研究では、途中式を書くため細かい理解度が判る。

人に教える行為や添削に関する研究として、以下の研究があげられる。澤田石ら [6] の研究では、学習者が生徒に教えるという LBT(Learning by Teaching) システムを構築し、学習者は生徒に教えることによって階層構造がある問題を理解できる。階層構造の問題の分割を行うための知識が正しく適用できていれば、さらに分割が起こり、分割しなくなったら終了である。分割を行うための知識が正しくなかったら、仮想生徒は教えてもらった知識では問題を解けないことをユーザに伝える。そうすることによって自身の誤りに気づくことができる。

Leelawong ら [7] の研究では、Betty システムというシステムを用いている。Betty システムは、河川生態系について学ぶ中学校の学生を対象としている。仮想生徒 Betty に河川生態系について教え、Betty をサイエンスクラブに入れる事が目的である。目的をユーザに持たせて仮想生徒に教えるタイプである。

緒方ら [8] の研究がある。緒方らは、電子メールの内容についての添削支援ソフトを開発した。例として、ある人 (A とする) が e-メールを書いた。それを他の人 (B とする) が添削をする。他の人の添削 (C とする) の影響を受けて、A に、添削されたものを見せる。ただし、このシステムの問題は、正解がないため、誤った添削をした人が多くいると、誤った結果を出してしまうことである。我々の研究だと、正解パターンが存在するためこのような問題は起きない。

## 3. 仮想生徒システム

仮想生徒は一種の人工知能であり、プロダクションルール [9] として動作する。仮想生徒の解答を自動的に生成するシステムについて説明する。問題として与えられた式とユーザの理解度をワーキングメモリにある状態とし、式変更を行う if-then ルール集合から適用可能なルールを抽出する。複数のルールが抽出される場合がある。その時競合解消戦略を行う。競合解消戦略として、ユーザの理解度を用いる。ユーザの理解度として正答確率、誤答確率を用いる。正答確率は、ユーザの添削式が対応している正しい式であった時の確率。誤答確率は、ユーザの添削式が対応している間違い式であった時の確率。正答確率、誤答確率を用いた理由として、一度のユーザの解答ではユーザの理解度は測れないと考えられるため、何度も繰り返し添削することによって実際のユーザの理解度に近づくことができるためである。仮想生徒の次の式を生成するための競合解消戦略は、結論部についての理解フラグが立っている場合には正のルールを、立っていない場合には誤のルールのいずれかを確率的に選択する。正しいルールに対になる形で誤ったルールが存在する。このことにより、ユーザが十分に理解している所については、添削を省略し、理解が不十分な所については、誤った解答により添削を促すことができる。さらに、仮想生徒の誤りを正しく添削する/誤って添削をする。もしくは、正しい式を添削しない/してしまうというパターンにより、細かい理解度を図ることができる。

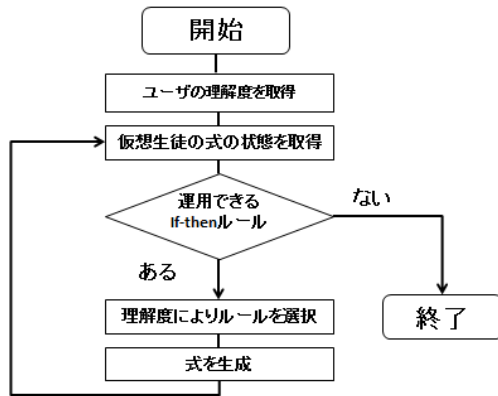


図2 仮想生徒システムの処理

仮想生徒の解答自動生成

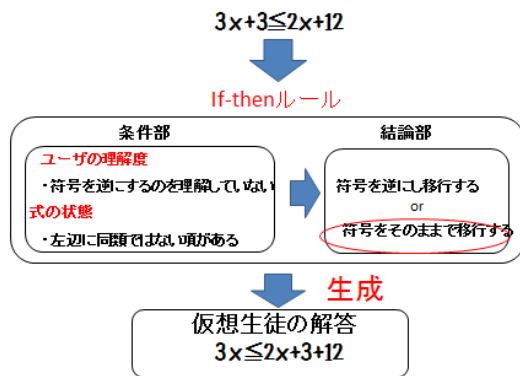


図3 仮想生徒の解答自動生成

表2 MathMLのタグの種類

タグ	説明
mo	演算子を中身に持つ
mi	文字を中身に持つ
mn	数字を中身に持つ
mrow	数式をグループにする

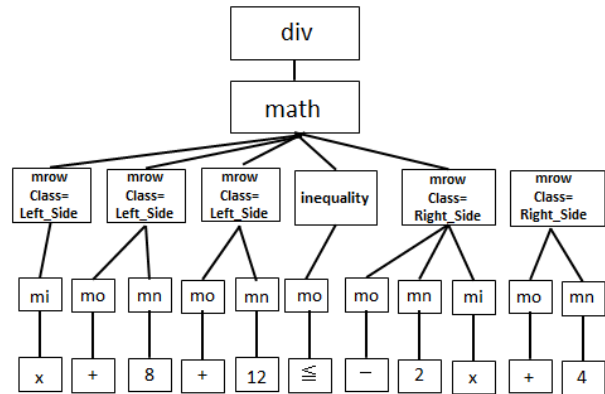


図4 MathMLの構造

ルールを追加すると、より深い範囲を学習をすることができるようになる。図3を例に説明する。if-thenルールを適用する式は、 $3x + 3 \leq 2x + 12$ である。ユーザの理解度は、“符号を逆にするのを理解していない”とする。式の状態は、“左辺にx類ではない項がある”である。この状態で適用可能なif-thenルールは、表1中のルール2と4である。

このif-thenルールを適用すると  $3x \leq 2x + 3 + 12$  と  $3x \leq 2x - 3 + 12$  が競合集合となり、2が選ばれる。それにより、 $3x \leq 2x + 3 + 12$  が生成される。ある単元に関する正解の式変換および、それに対となる不正解の式変換のif-thenルールを網羅することにより、正答もしくは誤答に到達するまで仮想生徒は、解答をし続ける。

## 4. 数式の変形方式

### 4.1 数式の構造化

本研究のシステムは、WEB上で動作するシステムである。数式を表示させるためにMathMLを用いる。MathMLは、数式を記述するためのマークアップ言語である。MathMLで良く使われるタグを表2で表した。理解箇所判定手法をわかりやすく説明するために木の形で表現する。

本稿では、理解箇所判定手法を行いやすくするために、MathMLの構造を以下のようにしている。目的である理解箇所判定手法を行うために、数式の項が左辺なのか、右辺なのか、不等号なのかを区別出来るようにする。数式の項が区別できるように、数式のグループを表すmrowタグを付けた。

左辺、不等号、右辺なのかは、mrowタグのclass名をLeft\_Side, Inequality, Right\_Sideとすることで区別する。

### 4.2 数式の変形

式変形は、新たな式を作るために必要である。式変形は、

図2は、アルゴリズムのフローチャートである。以下に仮想生徒の解答式の生成についてのアルゴリズムを示す。

- (1) ユーザの理解度配列を取得
- (2) 仮想生徒の式で一番新しく生成された式の状態を取得し、その要素を式の状態の配列に入れる
- (3) (1), (2)で取得した配列の要素で適用できるif-thenルールがあるならば、適用し式を生成する適用できないif-thenルールがないならば、仮想生徒はこれ以上式を生成出来ないということで終了する

表1のif-thenルールは、ある順序にしたがって解く場合についての不等式の単元を網羅している。表中の正誤列は正であれば正しい変形ルール、誤であれば誤った変形ルールを示している。このルール集合は、左辺にx類、右辺には数字類をまとめると言う形に限定している。また右辺に数字類が一つの場合でなおかつ左辺にx類が一つの場合のみしか割ることができない。これらの制約は、ルールを増やすことで、右辺にx類、左辺に数字類をまとめるようにすることができるようにすることも可能である。他にもx類や数字類が複数あった場合も、割ることができるようにすることもできる。さらに、四則演算や小学生の問題のif-thenルールを増やすと仮想生徒は細かいレベルでの式変換および間違いをすることが可能になる。他にも連立不等式や絶対値の付いた不等式、微分などの他の範囲のif-then

表 1 if-then ルール一覧

番号	条件部	結論部	正誤
1	移項され同類になった	同類を足す	正
2	(左辺に x 類ではない項がある)	符号を逆に移行する	正
3	(右辺に数字類ではない項がある)	符号を逆に移行する	正
4	(左辺に x 類ではない項がある)	符号を変えずに移項する	誤
5	(右辺に数字類ではない項がある)	符号を変えずに移項する	誤
6	((左辺に x 類)^(ひとつしか項がない)^(左辺の符号がプラス)^(右辺に数字)^(ひとつしか項がない))	(不等号が変わらず右辺の数字を左辺の数字で割る)	誤
7	((左辺に x 類)^(ひとつしか項がない)^(左辺の符号が-)^(右辺に数字類)^(ひとつしか項がない))	(不等号が逆になり右辺の数字を左辺の数字で割る)	正
8	((左辺に x 類)^(ひとつしか項がない)^(左辺の符号が-)^(右辺に数字類)^(ひとつしか項がない))	(不等号が変わらず右辺の数字を左辺の数字で割る)	誤

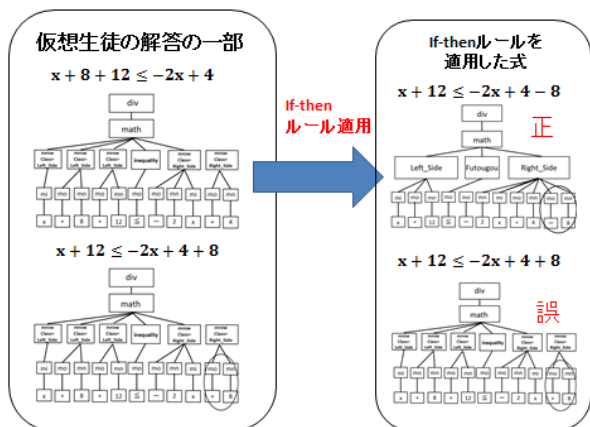


図 5 if-then ルール適用した式の木 (丸が適用された所)

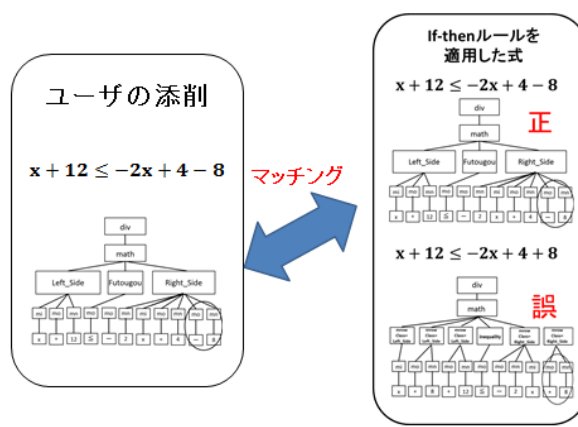


図 6 理解箇所判定手法

MathML の木の枝の親を変えることや、葉ノードを変えることで実現する。

$x + 8 + 12 \leq -2x + 4$  を例に if-then ルールによる変形例を示す。式の状態は“左辺に x 類ではない項がある”である。又、ユーザは、“符号を逆にするのを理解していない”とする。この時、表 1 のルール 2 および 4 が発火可能である。ルール 2、およびルール 4 の結論部の動作を説明する。このルールは正誤の対になっているルールであり、葉ノードの変更以外は同じ動作である。まず、以下の手順で部分木の親を変更し、式の移行を実現する。

- (1) 対象の mrow タグごとに mi タグがないかを確認する。なければ、3 に進む。あれば 2 に進む。
- (2) 次の mrow タグに進み、1 に戻る。進む mrow タグがなければ、5 に進む。
- (3) mrow タグの Class に Right\_Side と入れ、5 に進む
- (4) Right\_Side に入れ替える mrow タグの中の mo タグの中身の符号を変える。5 に進む
- (5) 終了する。

4 つ目の手順はルール 2 のみで適応し、ルール 4 では使わない。

## 5. 理解箇所判定手法

ユーザの理解箇所を判定するためにまずは、ユーザによる添削結果を解析する必要がある。本稿では、ユーザは、TeX 形式で添削結果の数式を入力するものとする。入力された数式を MathML 形式に変更することで、仮想生徒の式および、正解の式との比較を可能にする。なお、本稿では、TeX 形式で入

力を行っているが、直接的に仮想生徒の式を編集できるインタフェースによる入力なども考えられる。

理解箇所判定手法として以下の手法で行う。

(1) 添削対象となる仮想生徒が解答した式の直前の式に適用可能な if-then ルールをすべて適用して、式を複数作り、集合の要素とする。

(2) ユーザの添削した式と 1. で作った式を比較し、合致する式を生成したルールを抽出する。そのルールの結論部の式変形について理解・不理解を判定できる。

式の合致にはいくつかのパターンが考えられる。

- (1) 仮想生徒の誤りを正しく添削する。
- (2) 仮想生徒の誤りを間違って添削する。
- (3) 仮想生徒の正しい式を正しいと判断できる。
- (4) 仮想生徒の正しい式を誤って添削をする。

このうち、1, 3, 2, 4 の順に理解度は高いと考えられる。本稿では、単純に 1, 3 を理解。2, 4 を不理解としている。

図 6 を例に理解箇所判定手法の手順を説明する。ユーザの式  $x + 12 \leq -2x + 4 - 8$  は、仮想生徒の式  $x + 12 \leq -2x + 4 + 8$  を添削した式である。仮想生徒の式  $x + 12 \leq -2x + 4 + 8$  の変形される前の式である  $x + 8 + 12 \leq -2x + 4$  に if-then ルールを適用する。適用できるルールは、表 1 の 2, 4 である。2 と 4 を適用することにより、 $x + 12 \leq -2x + 4 - 8$  と  $x + 12 \leq -2x + 4 + 8$  が生成される。表 1 のルール 2 により生成された式と合致した。表 1 のルール 2 の結論部の式変形を理解したと判断できる。理解箇所判定手法によって、理解・不理解箇所が判り、その部分の答えをフィードバックすることにより、ユーザの理解度が上がると期待できる。

表 3 実験の状況

	A	B	C	D
1 回目	X/提案システム	X/ドリル	Y/提案システム	Y/ドリル
2 回目	Y/ドリル	Y/提案システム	X/ドリル	X/提案システム

今後、理解箇所判定手法で得た理解・不理解箇所から演習問題個人化を行う予定である。演習問題個人化として、ユーザの理解度集合から一番近くて、ユーザが理解していない知識を含んでいる問題集合を作る。下の式はそれを表している。 $p(x)$  は、 $x$  の知識を持つ問題を返す関数である。 $X$  は、問題に含まれている数学の知識の集合族であり  $x$  はある問題の知識集合である。 $Y$  はユーザの理解している数学知識の集合である。

$$C = \{p(x) \mid \max_{x \in X} (|x \cup Y|)\} \quad (1)$$

ユーザの理解度が高いルールの結論部を、理解している数学知識の集合として上記の条件に合う問題をユーザに提示する。

## 6. 実験

### 6.1 実験内容

実験の目的として、仮想生徒システムを行うことに添削の効果が得られるかを確認する実験を行った。ここでは、問題に対して取り組む姿勢を評価する。提案システムが普通に問題を解く(ドリル式)よりどのような点が違うのかを考察する。被験者は、普通に問題を解く方法と仮想生徒システムを使う方で問題に取り組んだ。その時被験者には脳波計測器を装着してもらいその結果で集中の違いを測定する。解く順番と問題の解き方の組み合わせによって実験の状況を揃えるために表 3 のように行う。表 3 の見方を説明する。問題, X, Y はそれぞれ 5 問ずつの問題である。表 3 のドリルは、普通に問題を解く。表 3 の提案システムは、仮想生徒システムを用いて解く。表 3 の ‘/’ は、‘/’ の左の問題を ‘/’ の右のやり方で解くことを示している。実験は 20 代男性 4 人に実験を手伝っていただいた。問題を解いたのち以下のアンケートに解答してもらった。“数学は得意ですか”, “ドリル式, 添削システムを用いた時どちらの方が楽しかったですか”, “ドリル式, 提案システムを用いた時どちらの方法の方が反射的に解きましたか”, “ドリル式, 提案システムを用いた時, 問題の式についてどちらの方が深く考えることができましたか”, “提案システムで深く考えることができた人に質問です。どのようなことを考えましたか”, “提案システムで深く考えることが出来なかった人に質問です。” “考えることが出来なかった理由として, どのような理由があると思いますか”

### 6.2 実験結果と考察

被験者に着けた脳波計測器により、集中している回数を取得しました。結果を表 4 に示す。A, B, C, D は被験者である。A のドリル式の  $\phi$  記号は、集中している結果を最初から取得していないため、データとして使わなかった。実験結果から、提案システムの方が集中回数が多いことが判った。また、アンケート結果より、数学が得意な人は一名、どちらともいえないと答えた人は 3 人であった。ドリル式, 添削システム, 楽しい

表 4 被験者の集中回数

	ドリル式 (集中回数)	提案システム (集中回数)
A	$\phi$	26 回
B	19 回	22 回
C	9 回	27 回
D	14 回	22 回

$x+3 \leq -2x+9$

$x \leq +3-2x$ +9	
$x-2x \leq +3$ +9	
$x-2x \leq +12$	
$-1x \leq +12$	

図 7 実験で用いた提案システムの出力

と答えた被験者は 3 人であった。ドリル式の方が反射的に解いたと答えた被験者は 4 人中 3 人だった。提案システムの方が深く考えることができると答えた被験者は全員だった。そう答えた被験者に“提案システムで深く考えることができた人に質問です。どのようなことを考えましたか”とアンケートで質問した所、以下の回答を得た、普段やらない入力方式だったのでとも頭を使った。よくも悪くも過程をしっかりと解いていくので、式のどこで間違ったか確認しやすいと思った。式が正しいかどうか、どこを間違えているか慎重になれた。これらのことより、添削をすることは集中することができると考えられる。なお、図 7 のような開発者が意図していないバグが発生した。図 7 の左は仮想生徒の途中式、右はユーザが添削するフォームである。本来は、左の仮想生徒の式の最後の左辺が  $x$  だけの形であり右辺が数字類しかない状態が出力されるべきであるがそのようになっていない。これはプログラムのバグによるものであり、仮想生徒により生成されるべき式でなかった。そのようなバグが発生したときの集中は数十秒間続いた<sup>(注4)</sup>。ユーザが意図していない、ミス(誤答)をわざと起こすことにより、学習になるのではないかと考えられる。

## 7. まとめと今後の課題

添削をするメリットである、知識を再確認できるというメリットを有効活用してもらえるために、仮想生徒システム、理解箇所判定手法を提案した。実験として、脳波計測器を用いて集中を測ることやアンケートを行った。そのことにより提案システムが深く問題を考えることができると判った。今後の課題としては、他の範囲や細かい理解度を測るために If-then ルールを用意する必要がある。ユーザの弱点を克服する最低限の if-then ルールを決定する方法も必要である。ユーザの理解度を用いて、演習問題個人化を行い、ユーザにとって最適な問題を出せるようにする。実際の添削では、式と言葉を用いるため、

(注4) : 表 4 には図 7 で起きたバグによる集中は回数に含んでいない

e-learning で添削を行う方法としても、式だけではなく言葉も用いる必要がある。

## 謝 辞

本研究の一部は、平成 25 年度科研費若手研究 (B)(課題番号：24700098) によるものです。ここに記して謝意を表すものとします。

## 文 献

- [1] 白井詩沙香, 仲村裕子, 福井哲夫, 数式入力が容易なドリル型数学学習システム「DigitalWork」の開発と評価, 情報処理学会研究報告. コンピュータと教育研究会報告 2013-CE-121(14), pp1-8, 2013.
- [2] 新竹 由基, 倉山 めぐみ, 平嶋 宗, 協調的学習を指向した作問学習のゲーム化, 電子情報通信学会技術研究報告. ET, 教育工学 110(453), pp235-240, 2011
- [3] 中村 晃, KIT 数学ナビゲーションを活用したリンク・バック・ラーニング KIT progress : 工学教育研究 12, pp29-38, 2007
- [4] 岡崎 泰久, 永江 由依, 田中 久治, 渡辺 健次, タッチペンによる直接操作を活用した携帯ゲーム機による実験観察学習支援システムの開発, 情報科学技術フォーラム講演論文集 9(3), pp619-622, 2010,
- [5] 本田 竜広, 横田 壽, 神田 隆至, 学習者対応型ヒント機能を備えたソフトウェアの開発 : 微分積分の自学自習用システム工学・工業教育研究講演会講演論文集 pp108-109, 2008
- [6] 澤田石礼秀, 山本 景子, 倉本 到, 辻野 嘉宏, 水口充, 人に教える行為により階層構造を持つ問題の解法を学ぶための個人学習支援システム, 情報処理学会研究報告. コンピュータと教育研究会報告 pp1-8, 2011
- [7] Leelawong , K., G. Biswas, Designing learning by teaching agents: The Betty's Brain system, International Journal of Artificial Intelligence in Education , Vol. 18, pp181-208, 2008
- [8] 緒方 広明, 葉田 善章, 矢野 米雄, VCoCoA:VCCML を用いた非同期型協同添削支援システム, 情報処理学会論文誌 40(11), pp3924-3933, 1999
- [9] Newell, A., "Production Systems: Models of Control Structures," in W. G. Chase, ed., Visual Information Processing, New York: Academic Press, pp463-526 1973.