形式論理のための教育訓練ツール（Mizar-MSE）のユーザインターフェース向上

職業能力開発総合大学校東京校　　福　良　博　史

Improvement of user interface with an instruction and training tool for Formal Logic tool Mizar-MSE

Hirofumi FUKURA

Summary

The purpose of “education in formal logic” is to promote students’ proper understanding on the concept of formal methods using Object Constraint Language（OCL）. Mizar-MSE is an effective tool for the students to learn both propositional and predicate logic in formal logic. But this tool could be only operated under the Character-based User Interface (CUI). So, I upgraded it so that students could operate it under the Graphical User Interface (GUI). In my design, I adopted the Universal Design Principles.

1. はじめに

現在の社会システムにとってコンピュータは重要な基盤として位置づけられている。銀行の ATM、鉄道の自動改札機、各種家電製品、航空機の予約システムそして車など。特に車に至っては、数十から百近いマイコンが搭載されている。コンピュータはソフトウェアが備わらなければただの箱でしかない。ソフトウェアに関連するトラブルで社会問題化し、顕在化する事例が増えてきている。このためソフトウェアの信頼性を求める声[(1)]が非常に強くなっている。このような高い信頼性を維持したシステムを開発し、サービスを提供する上で、常に「このシステムは人命や財産を安全に保護できているか？」ということに配慮しなければ許されない時代になってきている。

ソフトウェアの設計・実装において、テストを 1 万回行ったとか、徹夜でテストをしたとかいったところで努力は買っても信頼性の保障にはならない。たとえばテストしなければいけない条件分類が正しくできており、その確認テストがなされているか、その保障をどのように得ているのかを正しく評価できて初めて信頼性の議論ができる。先ほど述べたような体力を消耗する検査に基づいて信頼性を向上させることは非現実的であり、現実の“ものつくり”の現場では耐えられない。現在の技術においてこのような問題に対処するための方法としての最良の選択肢は、形式手法（Formal Methods）と呼ばれている方法[(2)]が考えられる。

この形式手法の色々な考え方の一覧が掲載されているホームページ Formal Methods[(3)]を見ると、沢山の方法論がある。しかし一般的にはどの方法論も通常のプログラミング言語を覚えるような容易さは無い。高度な教育と訓練が欠かせない。とくに数学の形式的な証明の方法論を用いて、ソフトウェアの設計段階、検査段階などで利用し、信頼性を高めることが、この手法の大きな特徴である。この形式手法紹介のホームページには Mizar と Object Constraint Language（OCL）も紹介されている。

Mizar は数学の証明の検証用プルーフチェッカであるが、ソフトウェアの信頼性検証用にも用いられている。また現在一番啓蒙活動が盛んなオブジェクト指向による設計方法論の UML の一部として OCL という言語がある。この OCL 言語は、形式手法に則った記述方式を採用している。この OCL は、証明の推論を行うことが目的ではなく、システムの或るポジションやタイミングにおける、あるべき姿を形式的に条件記述によって定義するための言語としての位置づけのため、利用の仕方は比較的容易な言語と考えられる。この OCL 言語は、これからのソフトウェア技術者にとって必須の道具となる可能性があり、この言語を理解するためには記号論理の教育訓練が必須となる。

ここで利用している Mizar-MSE は、Mizar のミニチュア版で本格的な数学の証明用のツールとしては不向きであるが、通称 Baby Mizar と呼ばれ、形式論理の入門教育用のツールとして利用されている。

数年前から、形式論理教育についての提案を行い[4]、プルーフチェッカ Mizar-MSE を用いた形式論理の教育訓練を実施[5]してきた。今年は、Mizar-MSE を用いた教育訓練の 3 年目となる。Mizar-MSE は、コマンド入力による操作をおこなわなければならず、いわゆる Character-based User Interface（CUI）によるツール操作環境での実習を行っていた。今年度は、この改造を行い Graphical User Interface（GUI）によるツール操作環境を整え、実習を行うこととした。このツールの設計には、ユニバーサルデザインの考え方を取り入れた。Mizar-MSE の CUI 環境での操作方法のプロセスと GUI 化したツールを用いた操作方法のプロセスの利害得失を論じる。

2.　Mizar-MSE 操作方法のプロセス

2.1　CUI 環境におけるプロセス

CUI 環境の場合、Mizar-MSE とテキスト・エディタの二種類のツールの操作方法を覚える必要がある。この操作方法のプロセスを説明する。

（1）証明したいことを、何らかのテキスト・エディタ（図１）を用いて記述する

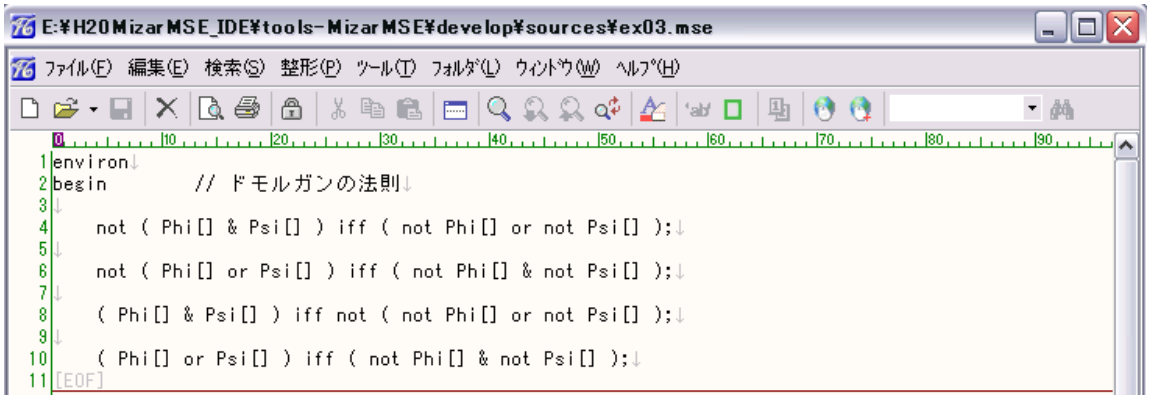

図１　テキスト・エディタでの証明の記述例

（2）コマンド・プロンプトを起動し、Mizar-MSE 用いた証明のチェック（図２）をする

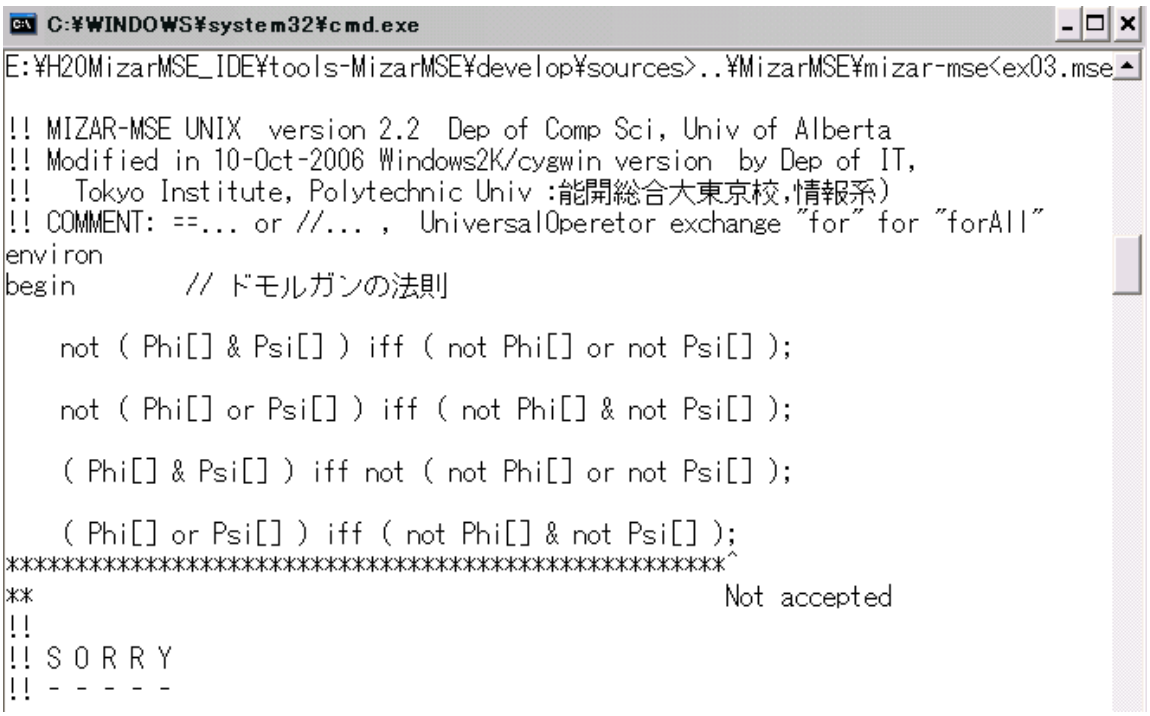

図２　Mizar-MSE による証明のチェック例（エラーあり）

（3）以上の結果証明に不都合があれば、（1）に戻りテキスト・エディタを利用し証明の記述を書き直し（図３）再度（2）のコマンド・プロンプトでの操作を行い、記述のチェックをする（図４）

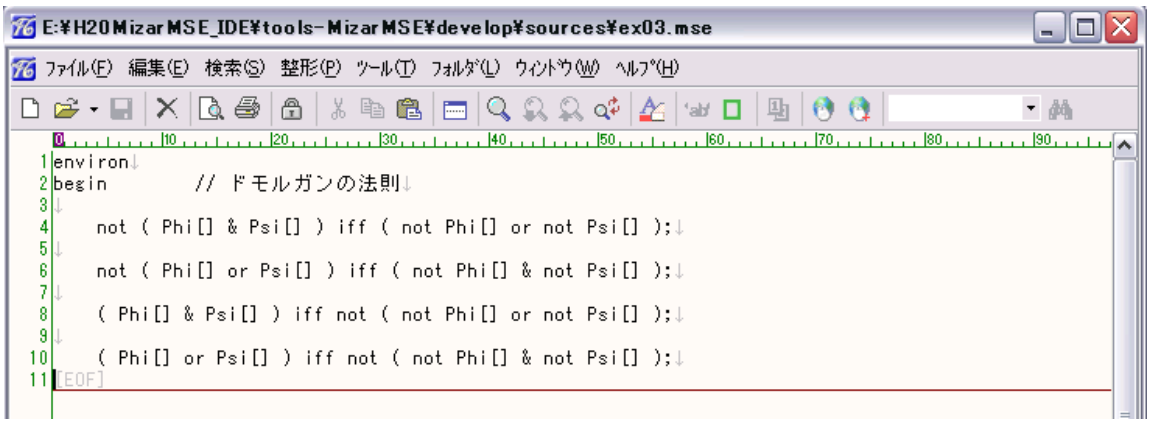

図３　テキスト・エディタでの証明の修正例

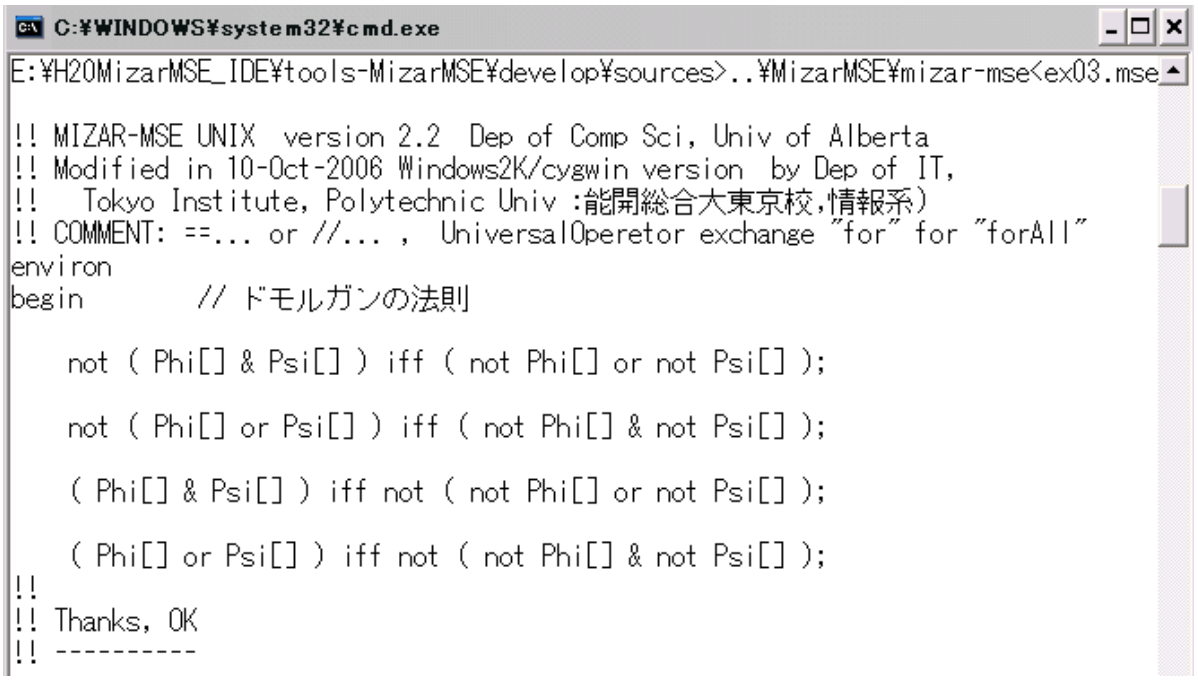

図４　Mizar-MSE による証明のチェック例（エラーなし）

以上のように複数のツールを使い分けて形式論理の教育訓練をしていくことになり、証明の内容に集中する前に CUI 環境での操作方法に慣れる必要がある。

2.2　GUI 環境におけるプロセス

GUI 環境は、基本的にツールを途中で交換しながら操作を行うということはしない。ワンストップサービスに配慮した。ソフトウェア開発などでよく使われている「統合開発環境」という表現を用いると、この場合のツールは、「統合形式証明（Mizar-MSE）環境」というような環境を提供することを目的とした。そのプロセスと機能の関係を以下に説明する。

（1）統合環境の立ち上げ

ツールを立ち上げる（図５）。

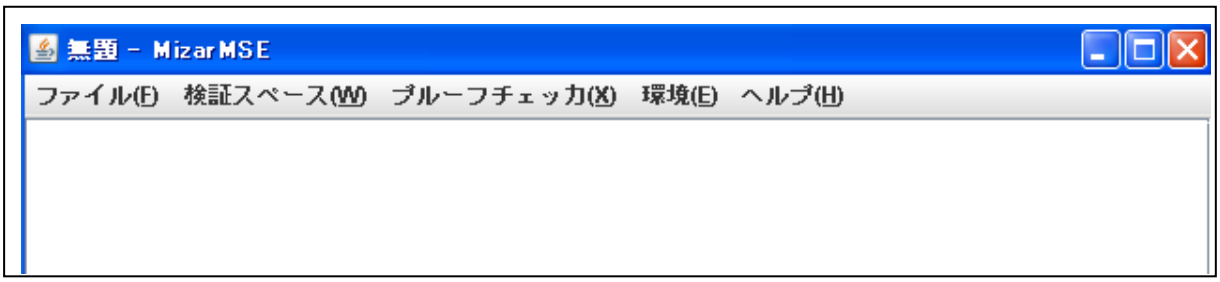

図５　Mizar-MSE 統合環境の初期画面

（2）テキスト・ファイルの選択

ファイル・メニュー（図６）から既存のファイルを修正するのかまたは新規作成するのかを選択する。そして編集が終わると保存する。

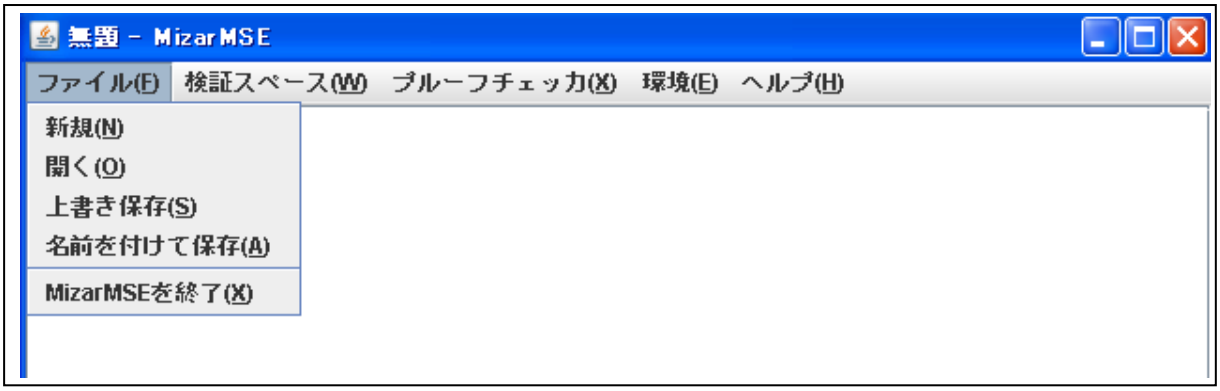

図６　Mizar-MSE 統合環境のファイル・メニュー

（3）検証中のテキストの再読み込み・修正保存

ファイル・メニューで選択したコードを検証用に用いる。このため検証用のコードのファイル操作を、ファイル名を指定せずに、更新が簡単にできるように、検証スペースというメニューを設けた。検証スペース・メニュー（図７）で入出力の操作を簡便に行うことができる。

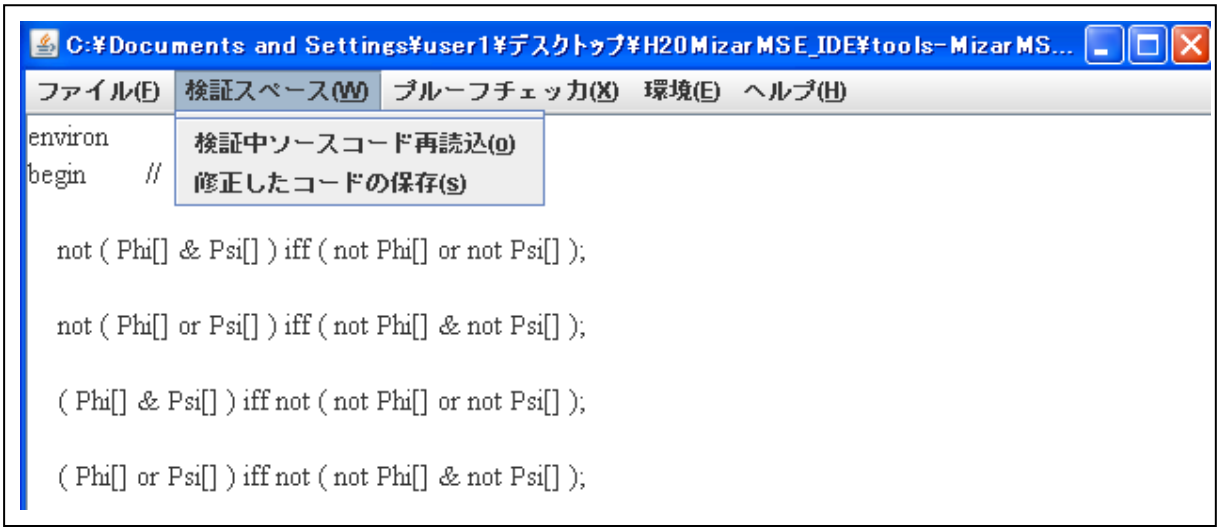

図７　Mizar-MSE 統合環境の検証スペース・メニュー

（4）プルーフチェッカを動かす

編集中のコードの証明の手順の正しさをプルーフチェッカ・メニュー（図８）の Mizar-MSE 実行により検証することができる。

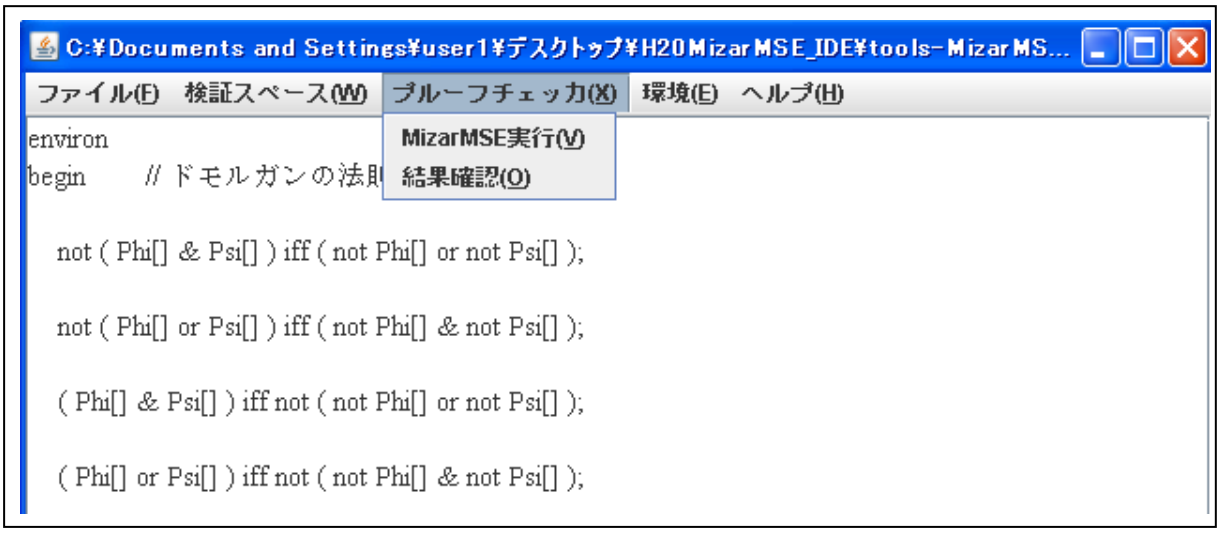

図８　Mizar-MSE 統合環境のプルーフチェッカ・メニュー

（5）プルーフチェッカによる検証結果の表示

プルーフチェッカによる形式証明の結果（図９）は以下のように統合環境の画面に表示される。

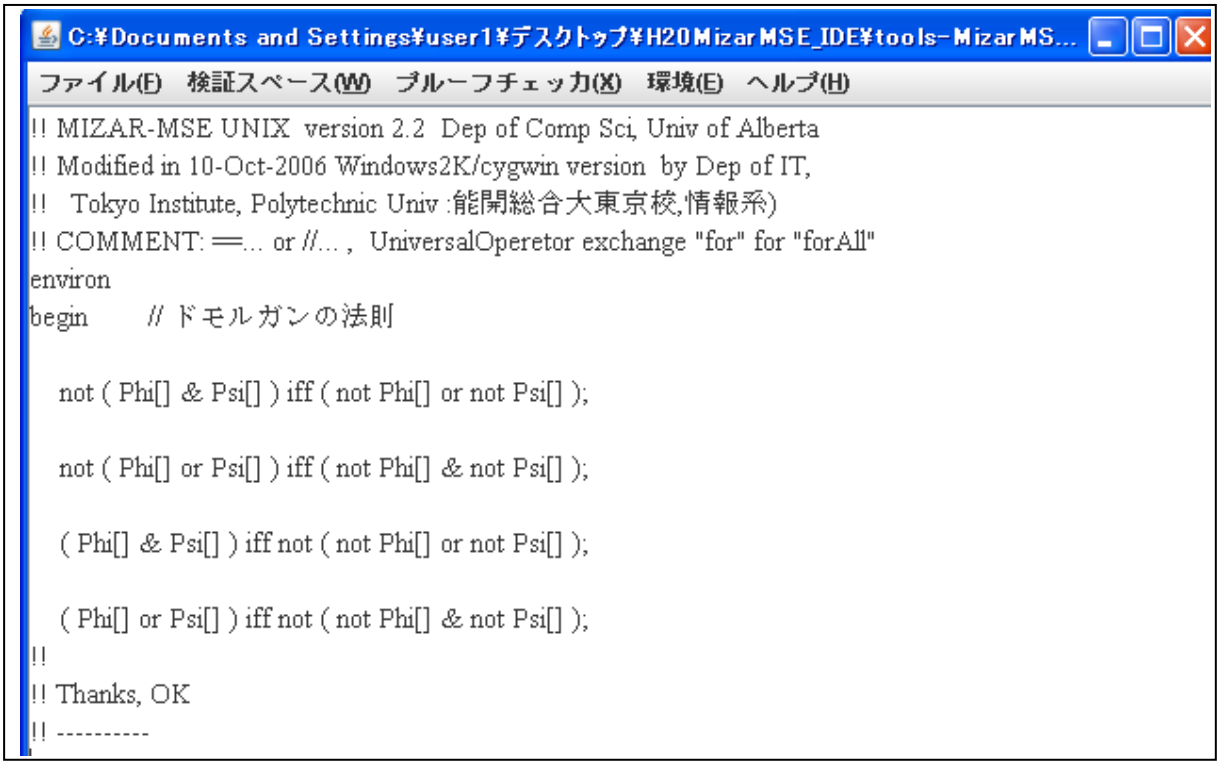

図９　Mizar-MSE 統合環境のプルーフチェッカ実行結果例

（６）プルーフチェッカの環境設定

環境メニュー（図 10）にてフォントの大きさを、大中小の３種類選択できる。L はプレゼンテーション用、中は目の不自由な人のためにフォントを大きくしている。

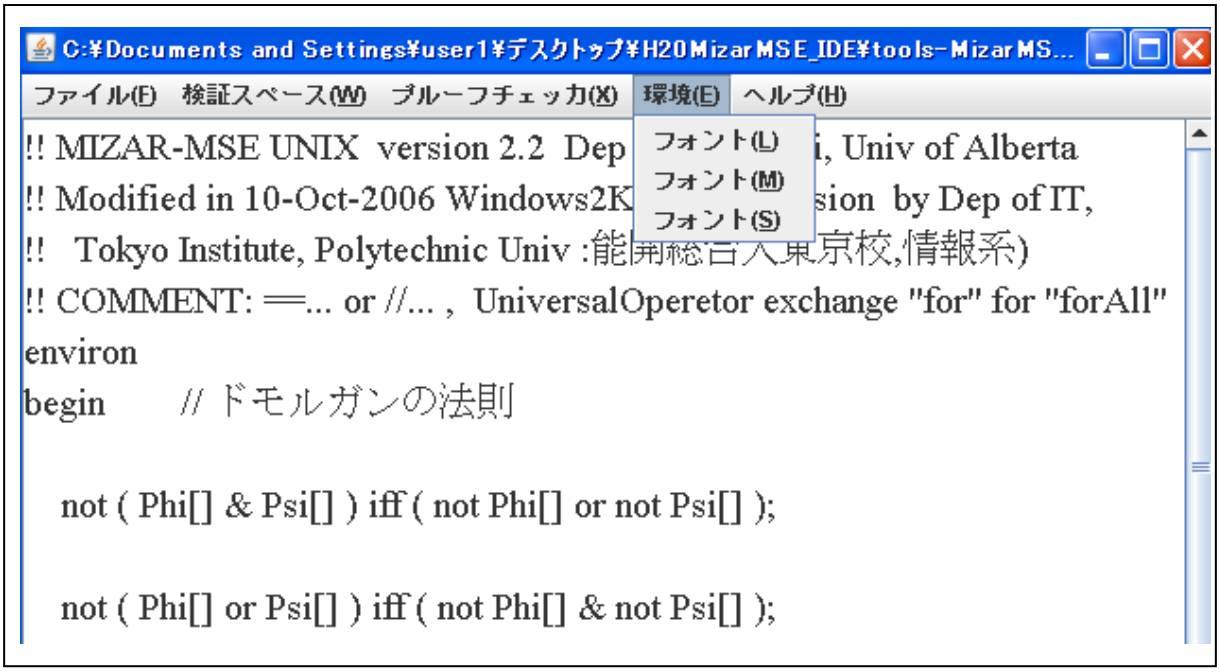

図 10　Mizar-MSE 統合環境の環境メニュー

（７）統合環境におけるインタラクション上での確認メッセージ

ツールを利用する上で操作に問題が生じた場合に警告メッセージ（図 11）などを表示し、利用者の注意を喚起する。

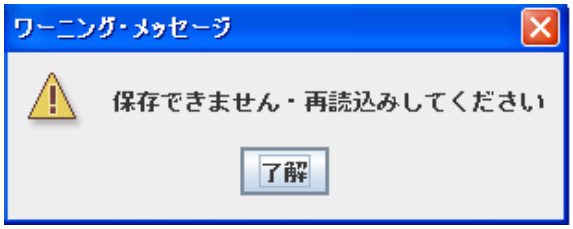

図 11　Mizar-MSE 統合環境の操作時の確認用メッセージ例

3. ツールの使い勝手に関する評価

ツールの使い勝手の一般論として、CUI と GUI の利害得失は、表１のように考えられる。

表１　CUI と GUI の一般的な利害得失

項番	CUI	GUI
1	コマンドを覚えておくことが必要	コマンドを覚えておく必要が無い
2	マニュアルが必須	マニュアルは必ずしも必須ではない
3	連続操作に向いている	連続操作には不向き
4	複雑な操作をバッチファイル化が可	バッチファイル化は一般に不可能
5	早く大量に処理可能	マウス操作などのため早さに限界
6	熟練者向き	一般向き

Mizar-MSE の場合は連続操作や早い操作性は必要ない。考えながら推論をコード化し、試行錯誤に徐々に証明を仕上げていく。この証明を仕上げていくプロセスを通して形式論理の訓練を行い、各自のペースで着実に習得していくことができればよい。このような教育訓練の考え方からは、操作方法に熟練し、操作の速度を早くする必要性は無い。学生が、証明の推論を考えることに集中し、証明のコードを構築していくことだけを考えればよい。そのためには GUI 方式のツールの方が使い勝手が良いと考えられる。

GUI 版を作成するために、基本的なデザインの考え方は、ユニバーサルデザインの７原則を評価尺度として採用し、設計に取り組んだ。試作版を制作した後、昨年 Mizar-MSE の CUI 環境でのツールを利用していた学生二人に、GUI 環境化したツールを使ってもらい、意見を求め改良を加えた。その改良版を用いて、使い勝手の調査を行った。

3.1　ユニバーサルデザインの７原則

ユニバーサルデザインの７原則[(6)]の概要を以下に示す。

（１）　第一原則：公平な利用
　　すべてのユーザに対して同じ方法で利用できる

（２）　第二原則：利用の柔軟性
　　ユーザのペースに合わせられる

（３）　第三原則：シンプルかつ直感的な利用
　　ユーザの経験、知識、言語能力などに依存せず理解しやすい

（４）　第四原則：情報の理解容易
　　ユーザの能力と無関係に効果的に情報を伝達できる
　　このため、冗長性のあるデザインが好ましい

（５）　第五原則：失敗に寛容
　　危険に対し、不利な結果を最小限に抑える

（６）　第六原則：少ない物理的努力
　　無理な操作を最小限にする

（７）　第七原則：接近と利用のためのサイズと空間
　　あらゆるユーザに重要な要素がハッキリ目に付き、快適に手が届く

3.2 ユニバーサルデザインへの適合性

Mizar-MSE の GUI 環境の統合ツールがユニバーサルデザインの観点から見たときにどのように適合しているのかを、各原則に照らし合わせてみる。

（１）第一原則（公平性）への適合性

すべてのユーザにとって GUI の起動により、誰もが同じ方法で各種メニューを開くことにより使えるようにデザインされている。

（２）第二原則（柔軟性）への適合性

エディタの利用、プルーフチェッカの利用その他各種機能は、メニューから選択することによりサービスが受けられる。このためユーザが自分の理解や早さのペースに合わせて自身でメニューを選択し行動をすることができるようにデザインされている。

（３）第三原則（シンプル）への適合

統合環境として、エディタもプルーフチェッカも一つのツールの中にメニューで選択して作業ができるようにシンプルなデザインを採用している。

（４）第四原則（理解容易）への適合

メニューを見ると何ができるのかが容易に理解できるようにデザインされている。

（５）第五原則（寛容）への適合

操作に異常が見受けられた場合は、即座に警告（たとえば図１１）を発するようにデザインされている。

（６）第六原則（操作極小）への適合

統合環境としたことで、いちいち別々のツールをその都度呼び出したりすることなく、メニューから選択することで必要な機能をすべて扱えるように操作性を極力減少させたデザインとなっている。

（７）第七原則（サイズ・空間）への適合

テキスト・エディタのフォントを３種類用意した。このことにより人それぞれの見易い文字の大きさを選択できる。またプレゼンテーション用に大きな文字を表示することが可能なデザインとなっている。

上記のようにこのツールを設計する上で、各原則にそれぞれ適合している機能を持つように、デザイン上の配慮をしておいた。しかし、逆にこのツール自身にこれらの原則から逸脱している操作上の問題点や機能があるのかという観点で考えると、その欠陥を発見するためには、設計者の独善に偏らないようにするため、設計者以外のユーザが利用して評価することが欠かせないと考える。

4. 学生による使い勝手の評価

昨年､CUI 版の Mizar-MSE を利用した学生のうち､ツールのことをよく覚えている学生８名に Mizar-MSE の GUI 環境版を利用してもらい、無記名でのアンケート調査を行った。

質問項目は、ユーザの使い勝手を中心にした。主にユニバーサルデザインの観点での質問を CUI と GUI のそれぞれについて５段階評価の選択式で問う形式を用いた。CUI と GUI の各々について７件の質問項目を用意した。この 14 個の質問のあと、質問 15 と質問 16 は、CUI と GUI のどちらが好ましいのかを比較して選択する設問とした。そして、最後に、記述形式で意見・感想を書いてもらうようにした。

アンケート集計の結果は、以下の通り。

質問１：CUI の手順（エディタで編集→コマンド立上→プルーフチェッカ起動）は簡単か？

一目で理解可	理解可	普通	理解困難	非常に理解困難
１人	１人	３人	２人	１人

質問２：GUI の手順（画面でのマウス操作）は簡単か？

一目で理解可	理解可	普通	理解困難	非常に理解困難
２人	３人	３人	０人	０人

質問３：CUI は理解しやすいか？（明確さ）

非常に簡単明快	簡単明快	普通	理解困難	非常に理解困難
１人	１人	２人	３人	１人

質問４：GUI は理解しやすいか？（明確さ）

非常に簡単明快	簡単明快	普通	理解困難	非常に理解困難
１人	４人	３人	０人	０人

質問５：CUI の操作系列は想像しやすいか（簡単）

非常に想像容易	想像容易	普通	想像困難	非常に想像困難
１人	１人	２人	３人	１人

質問６：GUI の操作系列は想像しやすいか（簡単）

非常に想像容易	想像容易	普通	想像困難	非常に想像困難
１人	４人	３人	０人	０人

質問７：CUI は長時間使っていて疲れやすいか？（持続性）

非常に疲れ難い	疲れ難い	普通	疲れ易い	非常に疲れ易い
０人	０人	３人	４人	１人

質問８：GUI は長時間使っていて疲れやすいか？（持続性）

非常に疲れ難い	疲れ難い	普通	疲れ易い	非常に疲れ易い
０人	０人	７人	１人	０人

質問９：CUI は事前に覚えておかなければいけないことが多いか？（自由度）

非常に少ない	少ない	普通	多い	非常に多い
０人	０人	１人	４人	３人

質問 10：GUI は事前に覚えておかなければいけないことが多いか？（自由度）

非常に少ない	少ない	普通	多い	非常に多い
０人	２人	５人	１人	０人

質問 11：CUI の操作は間違いやすいか？（安全性）

大変間違い難い	間違い難い	普通	間違い易い	大変間違い易い
０人	０人	１人	６人	１人

質問 12：GUI の操作は間違いやすいか？（安全性）

大変間違い難い	間違い難い	普通	間違い易い	大変間違い易い
０人	３人	３人	１人	１人

質問 13：CUI は思い通りに使えるか？（公平性）

非常に思い通り	思い通り	普通	思い通りでない	非常に思い通りでない
０人	０人	２人	６人	０人

質問 14：GUI は思い通りに使えるか？（公平性）

非常に思い通り	思い通り	普通	思い通りでない	非常に思い通りでない
０人	２人	５人	１人	０人

質問 15：今後利用するとしたら CUI と GUI のどちらを使いたいか？

大いに CUI	CUI	どちらでもよい	GUI	大いに GUI
０人	０人	０人	２人	６人

質問 16：CUI と GUI のどちらが覚えやすいか？

大いに CUI	CUI	どちらでもよい	GUI	大いに GUI
０人	０人	０人	２人	６人

質問１から質問１４までの各質問は左側が好意的意見、右側が否定的意見となるように五段階構成を用いている。そこで、五段階の評点に表２のように重みをつけ、式１を用いて評価した。

表２　質問１から質問１４までのアンケート項目の重み付け

＋２	＋１	０	－１	－２
X_1人	X_2人	X_3人	X_4人	X_5人

$$（＋２）＊X_1＋（＋１）＊X_2＋（０）＊X_3＋（－１）＊X_4＋（－２）＊X_5・・・・・(式１)$$

評価した結果を表３に示す。奇数番目の質問は、CUI について、偶数番目の質問は、GUI についての質問となっているので、それぞれ左右にわけて一覧できるように配置した。

表３　CUI と GUI のアンケートによる使い勝手の評価

質問主旨	CUI に好意的な意見		GUI に好意的な意見	
	質問項目	評価点	質問項目	評価点
手順	質問１	－１	質問２	＋７
明確さ	質問３	－２	質問４	＋６
簡単	質問５	－２	質問６	＋６
持続性	質問７	－６	質問８	－１
自由度	質問９	－10	質問１０	＋１
安全性	質問 11	－８	質問１２	０
公平性	質問 13	－６	質問１４	＋１

この結果から、CUI と GUI を比較すると、GUI 形式のツールが好ましいと答えている傾向が強いことが明確に判断できる。しかし、このなかで、長時間使っていて疲れないか、という持続性への問いに対しては、GUI が CUI より相対的には好ましい評価を受けているが、マイナス傾向にある。

質問 15 の CUI を使いたいと応えた人はいない。全員が GUI を支持している。質問１６の覚えやすさも同じように全員が GUI を支持している。

不足機能の洗い出しを行うために設定した「意見・感想」には二点の要望が書かれていた。

1. 第一の要望：配色分けされていたほうが見易い
   これは第四原則の理解の容易性のための冗長性を持たせるような要望と考える
2. 第二の要望：検証が完了したときにその旨のメッセージを表示して欲しい
   これも第四原則の理解の容易性のための冗長性を持たせるような要望と考える

5. まとめ

ユニバーサルデザインの観点で Mizar-MSE の GUI 環境のツールの機能設計を行い、かつ昨年 CUI 環境でこのツールを利用した学生に、今回の GUI 環境のツールを利用してもらう。そしてアンケートによる

使い勝手の調査を行った。その結果、Mizar-MSE の GUI 環境のツールの使い勝手は良好と判断できた。今年度からこのツールを用いて教育訓練を行っていく。

表２から、GUI の評価が CUI と比較した場合、相対的に高くなっているとはいえ、評価点をみると、ゼロとマイナスがある。安全性の観点は、GUI が評価されているとはいえ評価点はゼロとなっている。つまり積極的に良い評価とはなっていない。また持続性の観点は、CUI より GUI が評価を受けているとはいえ、評価点がマイナスとなっている。この点については、今後分析を深めていく必要がある。

アンケートの意見・感想の記述に冗長性への要望があがっており、今後はこのような冗長性での観点からの使い勝手の向上を図ることが望まれる。その他にも使い込んでいくと色々な要望が現れる可能性があるので、今後もユーザの意見を汲み取るようにしていきたい。

また、教育用であるから、ヘルプ機能も充実させることも今後の課題と考える。

〔参考文献〕

(1) http://sec.ipa.go.jp/koubo/2008/20080829.html
「情報システムの信頼性評価手法の調査」に係る公募
(2) 福良博史：”ソフトウェア技術者の道具としての「formal methods」”、茨城職業能力開発短期大学校　紀要、1998、pp.17～22
(3) http://vl.fmnet.info/ Formal Methods
(4) 福良博史：”情報技術に必要な記号論理の教育―UML その他の現在の設計手法との関係―”、職業能力開発報文誌、Vol.18 No.2 (2006)、pp.65～70
(5) 福良博史:”これからのオブジェクト指向設計に必要な形式論理の訓練の実践例―ミニ･プルーフ･チェッカ Mizar-MSE を用いて―”、職業能力開発研究、Vol.25 (2007)、pp.61～70
(6) http://www.design.ncsu.edu/cud/about_ud/udprinciples.htm
The Principles of Universal Design