



2009 年度上期未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当PM

夏野 剛 PM(慶應義塾大学 政策・メディア研究科 特別招聘教授)

2. 採択者氏名

チーフクリエイター: 岡田 健(慶應義塾大学 環境情報学部大岩研究室・非常勤講師)
コクリエイター : 秋山 優(慶應義塾大学 環境情報学部大岩研究室・非常勤講師)

3. プロジェクト管理組織

コシキ・バリューハブ株式会社

4. 委託金支払額

6,500,000 円

5. テーマ名

日本語プログラミング言語「言霊」の開発

6. 関連Webサイト

なし

7. テーマ概要

日本語プログラミング言語「言霊」を通じて、全ての人がプログラミングの概念を共有する社会を目指すことが、本提案の目的である。具体的には、プログラミングの概念を共有することで、以下のような社会を目指す。

- ・ 情報化社会を支える技術の根幹を社会全体で共有する

- ・自分が作りたいソフトウェアを自分でつくることができる
- ・自分が欲しいソフトウェアが何かを正確に伝えて、ソフトウェア開発者に作ってもらうことができる

このような社会を目指すために、我々日本人が日常的に使っている日本語をプログラミング言語として利用する。日本語プログラミング言語に着目する理由は、以下の通りである。

- ・日本語の語順「何を、どうする」は、印欧語の語順「どうする、何を」より、コンピュータに対する命令として優れている
- ・日本人は日本語で思考している

本提案の開発目標は、全ての人が、日本語で考えたアルゴリズムをコンピュータに伝えることができる日本語プログラミング言語「言霊」を開発することである。

8. 採択理由

これまでも日本語で書くプログラミング言語の発想は存在したが、本提案では、実際に商用で使えるレベルのコマンド数や応用性をもつ日本語プログラミング言語を開発することにより、プログラミングそのものを非プログラマーに開放することを目的としている。

日本語によるプログラミングはデバッグの負荷を減ずることができ、また、効用の高いプログラムを短期間で開発できるというメリットがあり、本提案が実現した場合の社会的便益は大きいと判断する。

9. 開発目標

(ア) 日本語プログラミング言語「言霊」の言語仕様

開発する言語に存在する字句要素、構文要素、日本語プログラミングの概念等を定義する。

(イ) ソースコードエディタ

プログラムを記述するために必要な機能(ファイル管理、ソース編集、標準入出力、コンパイラ、デバッグ、実行等)を提供する。MacOS、Windows 上にて動作する。

(ウ) 「言霊」標準教材

「言霊」を用いて制御構造(順次、条件分岐、繰り返し)、変数、関数を学習するためのオンライン教材。

10. 進捗概要

中間成果報告までに、基本的な機能を実装し、デモ・システムを完成させ、実際の授業などでフィードバックを得られるようにした。また、最終成果報告までに、言語仕様およびエディタ仕様の文書化も含めて完成させ、学習効果も含めたシステムの評価を行った。

11. 成果

1) 「言霊」言語仕様

過去において日本語プログラミング言語に「なでしこ」(2004 年度未踏ソフトウェア創造事業(未踏ユース)にて「日本語プログラミング言語の開発」(PM: 筧捷彦、開発者: 山本峰章)として開発された)がある。「なでしこ」はおよそ 1100 もの多種多様な命令を用意して、ユーザはその中から必要な命令を選ぶことでプログラムを作成する。一方で、「言霊」は単純で少数の命令を組み合わせることで、小規模から大規模なソフトウェアを開発することが可能である。

仕様については、「日本語プログラミング言語「言霊」文法仕様書」として文書化した。

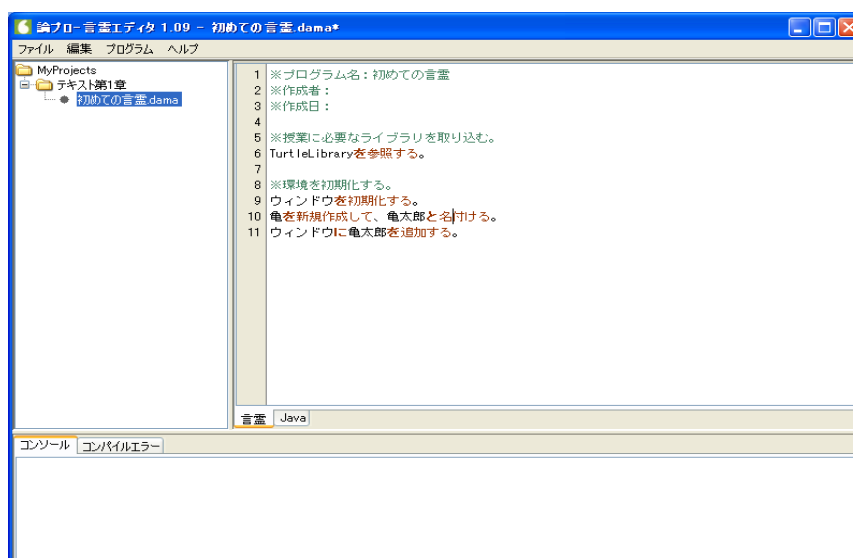
2) コンパイラ

コンパイラはソースコードエディタの一機能として提供される。

コンパイラの仕様については、「ソースコードエディタ仕様書」として文書化した。

3) ソースコードエディタ

ソースコードエディタのユーザーインターフェース画面は以下のようなものである:



エディタの仕様については、「ソースコードエディタ仕様書」として文書化した。

4) デバッガ

デバッガはソースコードエディタの一機能として提供される。

その仕様については、「ソースコードエディタ仕様書」の中で文書化した。

5) 「言霊」標準教材

本教材は、タートルグラフィクス環境を含む「言霊」標準ライブラリを用いて、プログラミングの基本概念である制御構造（順次実行、条件分岐、繰り返し）、変数、関数について学習することを目的とした。プログラムは全て日本語で記述されるため、初学者は細かな文法規則の習得に時間を費やすことなく、アルゴリズムの意味を考え、自分の力でアルゴリズムを構築する作業に集中することができる。本教材は、主に大学初年度の文系学生を対象とした演習型授業で用いられることを想定している。

当教材を用いて、言霊を教育現場で使うことでどのような効果が出るのか、効果測定を行った。詳細な測定方法と測定結果は、「プログラミング学習効果測定」として文書化した。

観察された効果を要約すると以下の通りである：

1. 言霊の方がソースコードを理解しやすい

シンプルなプログラム構造であれば言霊の方が命令文の意味を正しく理解できる可能性が高いと言える。

一方で、入れ子構造をはじめとする複雑なプログラミング構造は学習効果に差は現れなかった。講師による指導方法やプログラミング環境など、他の様々な方法を使って改善する必要がある。

2. 言霊の方がソースコード全体を読んでいる

Java クラスでは学生が問題となっているソースコード全体をちゃんと読んでおらず、言霊クラスではソースコード全体に目を通した上で問題を解いていた。

ソースコードを読むときの負荷が Java 言語と言霊に差があるために起こる現象と推察された。

言霊のように日本語で書かれたプログラム言語の場合は、ソースコードを読解するときの負荷がとても少ない。そのため記述されたプログラム全体に目を通しやすいと思われる。ただし、対象者がプログラミング初心者の時に限られる。

3. 言霊において理解しにくい記号があると、逆効果になる

これは、言霊を使うことによるマイナス効果であるが、言霊クラスは意味が理解できない記号に直面すると問題を解くことを諦める傾向が見られた。

これは言霊クラスの学生は日本語でプログラムを読むことに慣れているため、意味不明な記号である%演算子を見て、早々に問題に取り組むことを諦めてしまったと推察される。意味の分からない記号は読みたくないという反応が起こったと思われる。

この原因として、%演算子に関して言霊で、「～を～で割った余り」という日本

語表現を取り入れていれば回避できた問題と思われ、今後の改善点のひとつである。

12. プロジェクト評価

これまでも日本語で書くプログラミング言語の発想は存在したが、本プロジェクトでは、実際に商用で使えるレベルのコマンド数や応用性をもつ日本語プログラミング言語を開発することにより、プログラミングそのものを非プログラマーに開放することを目的としている。日本語によるプログラミングはデバッグの負荷を減ずることができ、また、効用の高いプログラムを短期間で開発できるというメリットがあり、本提案が実現した場合の社会的便益は大きいと判断する。岡田氏は長くこの言語の開発に携わっており、着実に前進してきた結果が本提案の成果に結びついたといえる。

13. 今後の課題

プログラミング言語は開発しただけではなく、実際に多くのユーザに利用されることを通して発展し、またその効果が広く社会に浸透していくべきである。本プロジェクト終了後、開発した「言霊」をより広く様々なユーザに使ってもらうために、「言霊」ユーザコミュニティの構築と運営、拡大が大きな課題となる。

また、本プロジェクトでは、中高大学生を対象とした日本語プログラミング教育を視野に入れ「言霊」の開発を行った。本プロジェクト終了後の展望は、プログラミング経験のない(あるいは十分でない)様々な分野のエキスパートを対象としたエキスパート・エンド・ユーザを育成することである。各分野のエキスパートに「言霊」を利用してもらうことによって、エキスパートが自分の専門分野で欲しいソフトウェアを自分で作成する、あるいは、プログラマと「言霊」を学んだエキスパートが要求分析、設計から実装までをシームレスに議論し、自分が欲しいソフトウェアをプログラマに発注できることが、今後期待される。

具体的な展望として以下の2点を挙げる。

1. 日本語プログラミング言語を用いた帳票作成

ユニバーサル・シェル・プログラミング研究所のユニケージ開発手法(IPAのソフトウェア・プロダクト・オブ・ザ・イヤー(R)2008を受賞)と組み合わせることで、業務用の帳票システムを、顧客側で開発することが出来るようになる。具体的には、中小企業や個人商店主でも、自分で帳票システムを開発することが出来る。

※販売データの個別集計

販売情報ファイルの商品ID・日付・個数を抽出して、
商品IDで商品部門情報ファイルを挿入連結して、
部門IDで部門名情報ファイルを挿入連結して、
日付を日付表記変更して、
商品ID・日付・曜日を削除して、
部門ID・部門名・週でソートして、
部門ID・部門名・週で個数を集計して、
部門ID・部門名が縦キー、週が横キーとして、マトリクス展開して、
販売実績集計ファイル(\$dir/ure2)に出力する。

日本語シェルスクリプトのコード例

2. 日本語プログラミング言語を用いた組み込み系システムの開発

組み込み系システムを扱う企業では、製品の動作に関する正当性をいかに保障するかが今後重大な課題である。しかし、現状では、部品製作や設計を行う機械担当者と、その動作を制御するソフトウェア製作者の間で、意思疎通が適切になされていないという問題が生じている。

この問題を解決するために、組み込み系システムを扱う企業が導入しているHCPチャートのような設計手法を拡張し、ソフトウェア開発における上流から下流までの全過程を日本語で行うべきとの提案を受けた。

組み込み系システムにおける実装は主にC言語によって行われているが、最下流の実行系表現以外はその抽象化である。このため、上流から下流までの全てを日本語に基づくことによって、機械担当者とプログラマは設計と実装の溝を認識するための議論を行いやすくなることが期待される。ただし、「言霊」が提供する日本語表現は、その分野で使われる言い回しや慣用表現を含み、その分野に特化された方言である必要がある。

今後、企業と協働し、組み込み系システム開発に特化した「言霊」の開発を目指した活動を行っていく予定である。