

# スマートスピーカーを用いたC言語ソースプログラムの 作成アシスタントシステムの開発

宇野 健・加藤 美沙・中川 樹

## Development of an Assistant System for Creating C Language Source Programs Using Smart Speakers

Takeshi UNO, Misa KATO and Itsuki NAKAGAWA

### 要 旨

本研究では、最近普及が著しいスマートスピーカーを用い、コーディング中に音声入力されたプログラミングに関する質問に対して、その回答をアプリ上に出力する、プログラミング初学者向けのC言語プログラム作成のためのアシスタントシステムの開発を行った。本システムはWebアプリケーションとして開発し、Webブラウザのみで動作可能としたため、パソコンやスマートフォンの機種やOSを選ばずに利用可能とした。これらにより、作成中の疑問の解決や、エラーへの対応を、キーボードから手を放すことなく行うことを可能とし、プログラムの作成に集中させることを目的とした。そして実際に本システムを用いてユーザーテストを複数回実施し、本システムの有効性や問題点を検証とともに、機能の改善を実施した。

### 1. はじめに

近年、Amazon EchoやGoogle Homeといった、スマートスピーカーやスマートフォンに搭載されているAIアシスタントの精度向上などを背景に、音声ユーザーインターフェースに注目が集まっている。操作する機器から離れていたり、両手が塞がっていたりする状況でも、音声だけで操作できるため、様々な分野に、広く応用されている<sup>[1]</sup>。

その中で、教育分野においてもスマートスピーカーの活用が始まりつつあり、初等教育向けとして英会話や九九のアプリなどが開発され、利用されている<sup>[2]</sup>。また、高等教育においても幾つかのスマートスピーカーの活用事例が報告されるようになってきた<sup>[3]</sup>。

一方、プログラミングの学習では、自宅等でのソースコード作成等の自学習が必要不可欠である。この際、学習者はプログラミングに関する用語や文法、サンプルコードを調べながら作成する。しかし、これまでの方法では、問題の解決のために、参考書やWebサイトなどを見る必要がある。そのためにキーボードから手を離すことが多くなり、結果的に集中力をそぐことになりやすい。プログラミングへのスマートスピーカーの活用事例は、障がい者向けでの利用については幾つか報告さ

れているが<sup>[4]</sup>、一般的なプログラミング教育向けのシステムは余り見られない。

そこで今回、音声入力によってコーディング中の疑問を解決するため、スマートスピーカーを用いた初学者向けのC言語プログラム作成のアシスタントシステムを開発した。これにより、プログラミングの学習者がキーボードから手を離さず、集中力を切らすことなく、ソースコード作成を行うことを可能にすることを目的とし、ユーザーテストによってその効果を検証した。

## 2. スマートスピーカーについて

### 2.1 スマートスピーカーの概要

スマートスピーカーは、端末から入力されたユーザーの音声を認識し、クラウド上でその意味を解析したのち、対応する命令を実行するものである。例えば、自宅でスマートスピーカーに話しかけるだけで、ネットショップで注文を行ったり、家電の操作を行ったりすることなどが可能となっている。2020年現在、国内で入手可能なスマートスピーカーは、Amazon Echo、Google Home、Clova WAVEなどがある。それぞれアマゾンジャパン、Google、LINE株式会社が提供しており、それぞれ提供されるAIアシスタント、クラウドサービス、音声認識結果の処理を行うエンドポイントと、それらの性能、そして利用料金が異なる。

本研究では、スマートスピーカーと連携するWebアプリ開発を行うため、そのプラットフォームが必要となる。Amazon EchoとGoogle Homeは独自のクラウドサービスを提供しているが、今回は、音声認識結果の処理に制限がないこと、認識結果が外部サーバから参照できること、音声処理に関する様々なAPIが用意されていることなどから、Google Homeを利用することとした。

### 2.2 C言語用語の音声認識実験

本システムの開発に先立ち、Google HomeにおけるC言語の用語の音声認識率等に関する調査を行うために、音声認識実験を行った。

#### (1) 実験概要

Google Homeでは、音声認識結果がクラウド上のGoogle Spread Sheetに文字情報として記録される。今回の実験では被験者にプログラムの用語をしゃべらせ、Google Spread Sheetに保存された情報と比較し、音声認識能力を調査した。今回の実験で用いたC言語の基本的な用語61個とした(Table 1)。被験者は5名であった(男性2名女性3名)。

#### (2) 実験結果

実験で用いた61個の用語のうち、正しく音声認識されたのは1人当たり平均35個であった。被験者すべてで正しく音声認識された用語は約半数の31個となった。正しく認識された用語には、日本語の用語が多かった。一方、正しく音声認識されなかった用語は、文字列操作関数strcatなどの読み方が統一されていないC言語特有の用語が多かった。その他に、%dはパーセントD、scanfはスキャンFのように、日本語とローマ字が混ざって認識された用語もあった。また認識結果のばらつきには、滑舌や声の大きさが関係していることが分かった。printfやstrcpyはカタカナで認識され、人により発音が変わるため、認識のパターンが一様ではなかった。それ以外にも「和」などの日本語でも、読みが一文字の単語では、誤認識されやすいことがわかった。

Table 1 音声認識実験に用いたC言語の用語

変数	ノートコール	文字列初期化	入力
変数宣言	かつ	日本語	getchar
データ型	または	ポインタ	putchar
型	アンドアンド	関数	strcpy
文字型	if文	関数呼び出し	文字列コピー
整数型	ifelse文	戻り値	strcpy
小数点型	for文	引数	文字列連結
char型	ループ	構造体	strlen
int型	while文	printf	文字列長さ
float型	配列	標準出力	乱数
和	配列宣言	出力	rand
差	配列初期化	%c	rand関数
積	2次元配列	%d	ASCIIコード
商	2次元配列宣言	%f	
等しい	2次元配列初期化	scanf	
等しくない	文字列	標準入力	

この実験結果より、スマートスピーカーを用いたC言語プログラム作成のアシスタントシステムを開発するに当たり、それぞれの用語に対応する、読み候補のデータベースを作成した。これらデータを実際の音声認識結果と比較することにより、複数通りの読みに対する用語の一致率を向上させることを試みた。

### 3. C言語プログラム作成のアシスタントシステムの開発

#### 3.1 システムの概要

本システムは学習者がより効率的にC言語プログラムを作成するためのアシスタントシステムである。本システムの利用の流れをFig.1に示す。

まず、学習者はパソコン画面にソースコード作成画面、ブラウザ上にアシスタントシステムを表示させておく（左ソースコード作成画面、右アシスタントシステム）。C言語のソースコードを作成中に、調べたい用語が出てきたとき（Fig.1中①）、学習者はGoogle Homeに対し、用語についての質問を話す（Fig.1中②）。クラウド上で音声認識処理が行われ（Fig.1中③）、アシスタントシステム上に、質問した用語の解説や用法のページが表示される（Fig.1中④）。

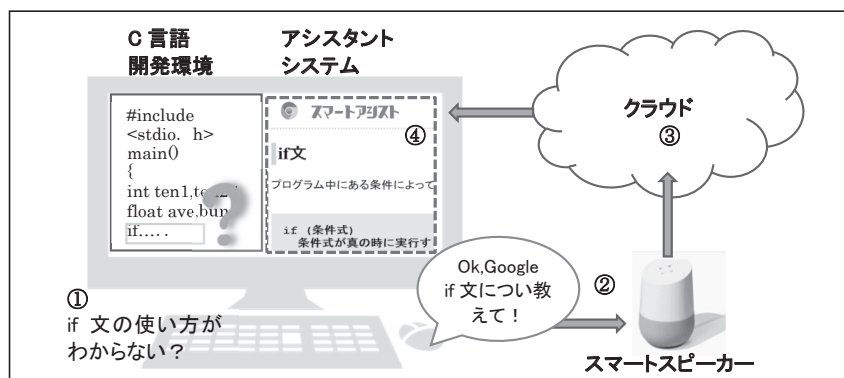


Fig.1 C言語プログラム作成アシスタントシステム利用の流れ

このように、学習者はソースコード作成中でもキーボードから手を離すことなく、質問に対する回答を得ることが可能となる。本システムの開発環境をTable2に示す。Webアプリケーションとして開発しているため、PCやスマートフォンの機種やOSを問わずに利用可能である。

Table 2 開発環境（アシスタントシステムサーバ）

OS	Microsoft Windows10 Professional
Web サーバ	Apache HTTP Server 2.4.20
開発言語	PHP 5.4.15 , Javascript, Python3, Google App Script
DBMS	MySQL 5.6
JS ライブラリ	jQuery3.3.1
Web サービス	Dialogflow, Google Sheets

### 3.2 システムの構成

本システムはスマートスピーカーに入力されたキーワードの音声認識結果を用いて対応する解説ページを表示させることである。スマートスピーカーから入力された音声の認識を行い、それをテキストデータ化する Googleサーバと、そのテキストデータを得て、対応する用語の解説ページを作成するアシスタントシステムサーバから成る。以下に、本システムの構成と、データの流れを示す (Fig.2)。

まず、Google Homeで音声入力した情報をGoogle Assistantで音声認識する (Fig.2中①)。様々なWebサービスを連携させる仕組みであるIFTTT<sup>[5]</sup>を用い、Google AssistantとGoogle Spread Sheetを連携させることにより、Google Assistantでの音声認識結果は、文字データとしてGoogle Spread Sheetに出力される (Fig.2中②)。ここまでの処理は、Googleのサーバ上で行われる。

次に、アシスタントシステムは、一定時間ごとにGoogle Spread Sheetに出力された文字情報をGoogleサーバから取得する (Fig.2中③)。ここで得た文字データについて、前章で作成した読み候

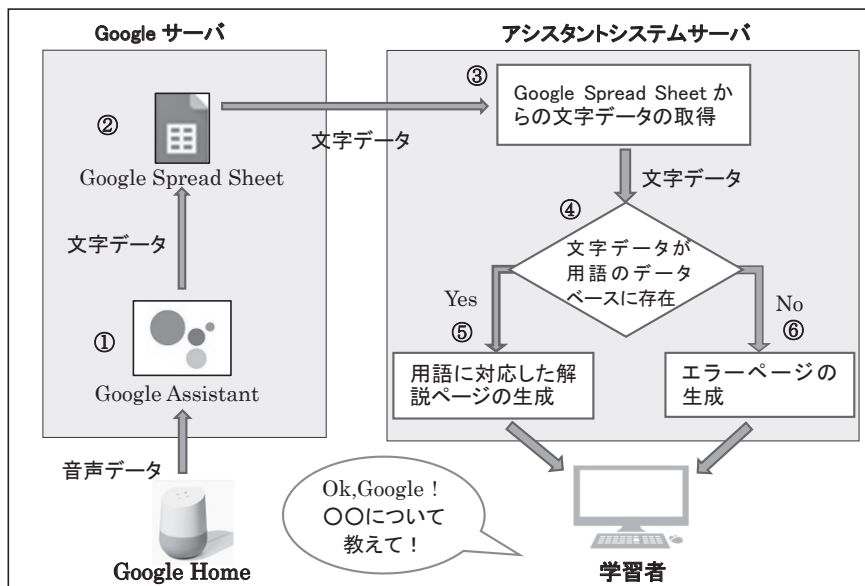


Fig.2 アシスタントシステムの機能とデータの流れ

補のデータベースへ照会を行い、(Fig.2中④)文字データがデータベースにあれば、対応した解説ページを生成し (Fig.2中⑤)、存在しなければ、エラーページを生成する (Fig.2中⑥)。解説ページには、その用語の意味や使い方、例文等、問題を自己解決するための情報を掲載した。

本システムでは、アシスタントシステムサーバから、GoogleサーバにあるGoogle Spread Sheetに対して一定時間ごと (3秒) に参照を行うように設定した。これにより、ほぼリアルタイムでの動作を可能とした。

## 4. ユーザーテストとシステムの改善

### 4.1 テスト概要

今回開発したアシスタントシステムの有効性を検証するため、ユーザーテストを3回実施した。まず、1回目のテストでは、システムの操作性と、インターフェースのチェックを主な目的として実施した。その結果を踏まえ、2回目では、1回目のテストによる改善と、音声認識精度向上のためにDialogflowを実装し、その効果の確認を主たる目的として実施した。3回目は2回目のテスト結果からの改善点の確認を行った。

### 4.2 第1回ユーザーテスト

プログラミング経験者の学生8名を対象に第1回目のユーザーテストを実施した。今回開発したアシスタントシステムとC言語の開発環境を用いて、指定した条件でC言語のソースコードを作成させた。実験後、被験者にインタビューを行った。

一人当たりの本アシスタントシステム使用回数は平均約13回で、用語に対応する解説ページ生成の成功率は約96%であった。生成が失敗した約4%のケース (エラーページが生成) は、入力された言葉が用語のデータベースになかったためである。

実験後のインタビューでは今までのC言語ソースコード作成との比較を重点に置き、意見を聞き取った。必要な情報を素早く得られたこと、キーボードから手を離すことなく情報を得られたこと、ネットで検索する時間が短縮された分ソースコード作成の時間が取れたことなどの肯定的な意見が多く挙げられた。一方で、音声入力用語をどのように聞けばよいのか迷う者が多く、また、聞いてからページ生成までにかかる時間が長いという意見もあった。

これらより、用語のデータベースの精度の向上と、文章や単語の羅列による音声入力の対応、ページ生成の時間の短縮等を行うことが課題であることが分かった。

そこで、1回目のユーザーテストの結果を元に、文章や単語の羅列による音声入力に対応させるため、WebサービスのDialogflowを利用した<sup>[6]</sup>。Dialogflowは自然言語の解析やAIとの対話の流れの構築が可能なサービスであり、認識結果とデータベースへ登録した語句との間で部分一致検索が可能である。そのため、用語や文章、単語羅列の一部を正常に認識できれば、その箇所を基に処理を行うことが可能になると考えられる。また、Dialogflowでは自動保存されたログから用語の学習が行えるため、対応語句を増やすことも容易になり、その結果、精度向上につなげることができると考えた。また、複数の候補がヒットする可能性がある用語の検索結果に対応させ、結果の絞り込みを可能とする検索結果ページの開発も行った。

### 4.3 第2回ユーザーテスト

第1回目のユーザーテストの結果から、システムの改善を実施し、2回目のユーザーテストを実施した。対象者はプログラミング経験者の学生3名で、1回目のテストの被験者は含まれない。実験方法は、1回目と同様である。

実験の結果、文章や単語の羅列での問い掛けに関しては、エラーが発生することはなかった。また、実施後のインタビューでは「Web検索に近い感覚で利用できた」という意見が得られた。

しかし、関数名の誤読、解説ページでの強調表現の過多による情報の探しにくさ、検索結果ページの構成などに改善すべき点が見つかった。例えば、被験者がscanf関数を「スカンプ」と誤読したため聞き返しが発生した。そこで、ログを参照し誤読内容を用語データベースに追加するようにした。また、解説内容に関数の振り仮名を追加し、学習者へ一般的な読み方の告知を行うように修正した。

また、追加実装に実装していた解説ページの強調表現では、問い掛けた用語に関する箇所のハイライトが過剰に機能し、見るべき情報が探しづらくなった。そのため、ハイライトの付く箇所が限定的になるように用語データベースを修正した。

今回の改善で追加した検索結果ページの候補提示機能において、候補が1つの場合も選択が必要となっていた。これに対して、音声認識機能、解説表示機能の両方に分岐処理を追加することで候補が1つの場合には即座に解説ページに遷移するように改善した。

### 4.4 第3回ユーザーテスト

2回目のテストで見つかった問題点の改善後、動作確認とシステムの有用性の検証を目的として二回目のテストを行った。対象はプログラミング経験者4名とし、第1回、第2回のユーザーテストの被験者は含まれていない。

実験の結果、音声認識エラーは一度も発生せず、認識精度の向上が確認できた。実験後のアンケート結果では、「候補が出るので目的のページにたどり着きやすい」「ハイライトがあり、見るべき場所が見つけやすい」という肯定的な意見が挙がり、今回行った解説表示ページの見やすさへの改善の効果が認められた。

## 5. まとめ

本研究では、C言語ソースプログラム作成の効率化を図ることを目的に、スマートスピーカーを用いたC言語ソースプログラム作成のアシスタントシステムの開発を行った。これに先立ち、Google HomeのC言語用語の認識の能力を調査し、用語のデータベースを作成した。それを用いてシステムを開発したことで、用語の読みの違いなどを考慮に入れた、音声入力によるアシスタントページの自動生成を可能とした。

本システムの有用性と改善点を確認するため、本学の学生を対象にユーザーテストを3回実施した。1回目の実験では、用語のデータベースに登録した用語については、正しく対応したページを生成することができた。実験後のアンケート調査では、キーボードから手を放すことなく知りたい情報をスムーズに得られる点で、効率的にソースプログラムの作成が可能となったという意見が得られた。しかし、文節での単語の誤認識や、文章での用語の認識ができない、解説ページの見やすさなどで問題を指摘する声が上がった。

この問題を解決するため、WebサービスのDialogflowをシステムに導入することで、音声認識の精度を向上や文章や単語羅列といった問い掛けに対応した。改善したシステムを用い、2回目、3回目のユーザーテストを実施した。その結果、本システムを用いたユーザーテストでは「プログラミング用語の認識精度が高い」や「単語羅列に対応しているためWeb検索に近い使い心地が得られた」という、本システムに対する肯定的な意見が得られた。

今後の課題として、よりあいまいな入力にも対応できるように用語のデータベースの精度の向上させる必要がある。また、文章や単語の羅列での音声入力に対応するとともに、より早く解説ページを表示させることが挙げられる。最終的には、人に質問するような音声入力での対応と、それに対する的確な回答をスムーズに与えられるよう、AIの利用などを検討していきたい。

本研究は、公益財団法人サタケ技術振興財団 平成30年度大学研究助成金により成された。

### 参考文献・URL

- [1] 里山南人, 一円 治, ステップバイステップで力がつく Google アシスタントアプリ開発入門, ソシム (2018)
- [2] Google Home, [https://store.google.com/jp/product/google\\_home\\_learn](https://store.google.com/jp/product/google_home_learn) (2020.9.10 アクセス)
- [3] 甲斐晶子, 松葉龍一, 合田美子, 鈴木克明, 受身形転換練習のためのスマートスピーカー (Alexa) 用機能の開発, 日本教育工学会, 第34回全国大会発表論文集 (2018)
- [4] 鶴見昌代, 宮城愛美, スマートスピーカーのスキル開発による視覚障害者のプログラミング教育プログラムの構築, 筑波技術大学テクノレポート, Vol.27, pp.117-118 (2019)
- [5] IFTTT Help Center, What is IFTTT?, <https://help.ifttt.com/hc/en-us/articles/115010325748-What-is-IFTTT->. (2020.9.10 アクセス)
- [6] Google Cloud, Dialogflowの基本 | Dialogflowのドキュメント, <https://cloud.google.com/dialogflow/docs/basics?hl=ja> (2020.9.10 アクセス)