情報処理学会論文誌　教育とコンピュータ　Vol.6　No.3　21–30 (Oct. 2020)

研究論文

スマートスピーカーを題材にした高等学校におけるプログラミング学習環境の提案

島袋 舞子[1,a)]　本多 佑希[1]　兼宗 進[1]

受付日 2019年11月23日，再受付日 2020年4月5日，
採録日 2020年6月27日

概要：新学習指導要領で高等学校の共通教科「情報」では，情報 I において人工知能（AI）や Web API などの外部のプログラムとの連携を含めたプログラミングが扱われる．我々はその題材として，スマートスピーカーに着目した．スマートスピーカーのアプリケーションを開発することで，音声認識の AI や Web API を体験することができる．そこで，Web ブラウザ上で動作するプログラム開発支援環境を開発した．本論文では，開発したプログラム開発支援環境の概要と，高等学校で行った授業を報告する．生徒へのアンケート結果と教員へのインタビューにより，高校生がスマートスピーカー専用のアプリを開発できることと，スマートスピーカーのアプリ開発をとおして生徒がスマートスピーカーの仕組みや Web API について学ぶことが可能であることを確認した．

キーワード：プログラミング教育，スマートスピーカー，Web API

A Programming Environment for Smart Speaker Applications

Maiko Shimabuku[1,a)]　Yuki Honda[1]　Susumu Kanemune[1]

Received: November 23, 2019, Revised: April 5, 2020,
Accepted: June 27, 2020

Abstract: In the new course of study, students will study programming that includes external programs such as artificial intelligence (AI) and Web API. So we propose programming learning environment for smart speakers that runs on a Web browser. In this paper, we report the lessons in a high school using the environment. By the results of questionnaire surveys to students and interviews with teachers, we confirmed that high school students can develop apps dedicated to smart speakers, and students can learn about smart speaker mechanisms and web APIs through smart speaker app development.

Keywords: programming education, smart speaker, Web API

1. はじめに

2020 年から初等中等段階でプログラミング教育が必修となる [1]．プログラミング教育が必修となった背景として人工知能（AI）などの第 4 次産業革命による社会や生活の変化があげられる．AI を活用したデバイスが多く登場しているが，その 1 つにスマートスピーカーがある．

スマートスピーカーは AI による音声認識技術により利用者が発した音声を解析し，特定の発話に応じてスキルと呼ばれる適切なアプリケーション（以下，アプリ）を起動・操作するデバイスである．音声認識を行う AI や起動するアプリはサーバ上に存在する．スマートスピーカーは発話を認識するとネットワークを介して AI やアプリとやりとりをする．スマートスピーカーで動作するアプリは開発元が提供しているソフトウェア開発キット（以下，SDK）を利用することで，自ら開発することも可能である．

高等学校の新学習指導要領における共通教科「情報」[1] では，外部ライブラリや AI を用いて問題を解決するためのプログラミングが扱われており，生徒がスマートスピーカー上で動作するアプリを開発することは題材として適し

[1] 大阪電気通信大学
Osaka Electro-Communication University, Neyagawa, Osaka 572–8530, Japan
[a)] shimabuku.m@gmail.com

ていると考える．

そこで本論文では，プログラミングを専門としない高校生が，スマートスピーカーのアプリ開発をとおして外部ライブラリを利用したプログラミングを体験するための教材の実現を目的とする．

2. スマートスピーカーと教育

2.1 スマートスピーカーの教育利用

スマートスピーカーは様々な学習の場面で活用されている．スマートスピーカーとの対話により日本語の受身形や擬音語を学ぶもの [2], [3]，算数の文章問題の学習を支援するもの [4] などが存在する．

アプリの開発をとおしてプログラミングの学習を行った実践も存在する．自作したスマートスピーカーで動作する英語翻訳などの Q & A アプリを Python と Julius で作成した高等専門学校の授業 [5] や視覚障害を持つ大学生に対して JavaScript（Node.js）によるスマートスピーカーのアプリケーション開発をプログラミング学習の一部として取り入れた授業 [6] がある．これらの実践はプログラミングを専門とした大学生や高専生を対象としており，専門的な開発環境を使用している．

2.2 高等学校におけるプログラミング教育

高等学校でプログラミングは，共通教科「情報」で扱われる．新学習指導要領において必履修科目となる情報 I では，主に「(3) コンピュータとプログラミング」でデータやデータ構造，外部のプログラムとの連携を含めたプログラミングが扱われる．外部のプログラムと連携したプログラミングについては，「画像認識や音声認識及び人工知能などの既存のライブラリを組み込んだり，API を用いたりすることなどが考えられる」[1] と例示されており，従来のようにすべてを自分で記述するのではなく，OS や言語，そしてインターネットで公開された外部のライブラリを適切に活用することで，問題解決につながる実用的なプログラミングを体験することを求めている．

一方，これまでに高校生に対してサーバに蓄積したデータを扱うプログラミング [7]，クライアントとサーバサイドのプログラミングを扱った実践 [8] や情報システムのしくみをデータベース操作のプログラミングの体験を通して学んだ実践 [9] などがあるが，インターネットで公開された Web API を利用することで，外部のライブラリを適切に活用した報告はほとんど存在しないため，授業実践の事例が求められている状況と考えられる．

2.3 スマートスピーカーを利用したプログラミング教育

本論文では，音声認識の AI や Web API を扱うプログラムの題材として，スマートスピーカーのアプリに着目した．スマートスピーカーのアプリを開発するためには，音声認識を行うためのライブラリを利用する．また，天気予報や電車の運行案内を知らせるアプリでは，Web API での問い合わせによりインターネット上で公開されているデータを取得し，そのデータを加工，利用している．一般的なスマートスピーカーのアプリの開発方法について，次章に記述する．

(1) 開発環境のアカウントを取得するか PC にインストールする
(2) 開発元のサーバーに情報を登録する
- 開発者 ID
- アプリが動作するサーバー，アプリ名など
- アプリで音声認識する単語

(3) アプリの機能をプログラミングで開発する
(4) 実機でテストする

図 1　スマートスピーカーの一般的な開発手順例
Fig. 1　A flow of developing smart speaker applications.

3. スマートスピーカーのアプリ開発

3.1 標準的な開発方法

スマートスピーカーのアプリは，開発元が提供している SDK を用いて開発を行う．アプリの開発手順は使用するスマートスピーカーによって異なるが，主に JavaScript（Node.js）や Python などのプログラミング言語を使用する．**図 1** に代表的な開発手順を示す．音声認識を行う AI には音声を認識する際に名詞を記述した辞書が必要となる．辞書への登録は，スマートスピーカーの開発元が提供しているサポートページで行う場合が多い．

スマートスピーカーのアプリ開発は国内外で行われている．アプリの開発は，スマートスピーカーの開発元が提供している SDK をダウンロードしてローカル環境で開発を行う方法 [10] や開発元が提供する環境を利用してオンライン上で開発を行う方法 [11], [12] が用いられている．**図 2** に LINE 社が提供する SDK [10] を用いて記述した例を示す．図 2 は「和食」と呼びかけられたら「寿司はいかが？」と返答し，「和食」と呼びかけられたら「オムライスはいかが？」と返答するプログラム例である．

提供される SDK を利用する場合，複数の命令呼び出しやイベント処理，分岐を組み合わせた複雑なプログラムを記述する必要がある．また，Web API を利用する場合は，JSON 形式や XML 形式でデータを取得し，それをプログラム中で扱えるように別途処理を行う必要がある．そのため，生徒には難易度が高く授業で扱うことが難しいという課題があった．

3.2 簡易的な開発環境

専門的なプログラミングの知識を必要としない開発環境も研究が進められている．スプレッドシートに対話を定義

することでアプリを開発する環境 [12] や用意されたアプリのテンプレートを埋めていく形の開発環境 [13], [14]，複数の状態に対して音声などの入力を表す線で結ぶことで遷移を記述する環境 [15], [16], [17] などが存在する．

BotTalk [18] は，構造化データを表す YAML をベースにしたマークアップ言語であり，データのやりとりを記述することでアプリを開発する．プログラミングの知識は必ずしも必要ないが，XML や JSON のように構造化されたテキスト形式で動作を記述する必要があり，初心者には事前の学習が必要になる．

これらの開発環境はアプリの開発だけを考えると便利だが，プログラミングの知識が求められないことから，プログラムを活用したアプリ開発の学習には必ずしも適していないと考えられる．

3.3　教育に適したアプリ開発環境の提案

通常の言語でも，煩雑な部分を隠蔽するライブラリを用意することで，ある程度簡略化した記述が可能になる．**図 3** は図 2 の定型的な部分をライブラリ化して隠蔽することで最大限に簡略したプログラム例である．図 2 よりは簡潔に記述することができるが，高校生にとって記述することは簡単ではない．

そこで我々は，Web ブラウザ上で動作し，プログラムの編集やサーバの用意を支援する環境を開発している [19], [20]．**図 4** にプログラム例を示す．この環境では入力語に応じた関数定義を記述することで，図 2 や図 3 のような文字列比較や条件分岐を記述せずに，プログラムを記述できる工夫を行った．また，Web API で取得した JSON 形式のデータをオブジェクトのプロパティによる木構造データの形で格納することで，「天気データ：wind:speed」のような簡潔な記述で扱えるようにした．

4.　スマートスピーカーのアプリ開発支援環境

4.1　概要

図 5 に，提案する開発環境のシステム構成図を示す．このプログラム開発支援環境は，Web ブラウザ上で動作し，プログラムの編集やサーバへのプログラムの登録を支援する．開発環境は開発者が提供するサーバから自動的にダウンロードされて実行されるため，端末への開発環境の事前のインストールは不要である．対応するスマートスピーカーとしては，現在は LINE Clova [21]（以下，Clova）用の開発環境に対応している．

開発環境を学校などの授業で利用する場合には，教員が自身の LINE アカウントを使い，LINE 社のサーバに授業 ID や授業で使用する単語を登録する．生徒は教員が登録した情報を利用してアプリ開発を行えるため，生徒の登録は不要である．

```
const clova=require('@line/clova-cek-sdk-nodejs');
const express=require('express');
const bodyParser=require('body-parser');
const clovaSkillHandler = clova.Client
  .configureSkill()
  .onLaunchRequest(responseHelper=>{
    responseHelper.setSimpleSpeech({
      lang:  'ja',
      type:  'PlainText',
      value:  ' カスタムスキルを起動しました',
    });
  }).onIntentRequest(async responseHelper=>{
    const intent=responseHelper.getIntentName();
    const slots=responseHelper.getSlots();
    if(intent=='customIntent'){
      if(slots['customSlot']==' 和食'){
        responseHelper.setSimpleSpeech(
          clova.SpeechBuilder
            .createSpeechText(' 寿司はいかが？')
        );
      }else if(slots['customSlot']==' 洋食'){
        responseHelper.setSimpleSpeech(
          clova.SpeechBuilder.
            createSpeechText(' オムライスはいかが？')
        );
      }
    }
  }
}).handle();
const clovaMiddleware = clova.Middleware({
  applicationId:  "xxx.yyy.zzz"
});
const app = new express();
app.post(
  '/clova',
  clovaMiddleware,
  clovaSkillHandler
);
```

図 2　提供される SDK で作成した JavaScript のプログラム例
Fig. 2　An example of JavaScript program using SDK.

```
const clova=require('./clova');

clova.listen('xxx.yyy.zzz', (intent,slots)=>{
  if(intent=='customIntent'){
    if(slots['customSlot']==' 和食'){
      clova.say(' 寿司はいかが？');
    }else if(slots['customSlot']==' 洋食'){
      clova.say(' オムライスはいかが？');
    }
  }
});
```

図 3　専用ライブラリで簡略化した JavaScript のプログラム例
Fig. 3　An example of program simplified by a library.

```
和食＝「クローバー！”寿司はいかが？” 話す」。
洋食＝「クローバー！”オムライスはいかが？” 話す」。
```

図 4　提案する開発環境でのプログラム例
Fig. 4　An example of program of this proposal.

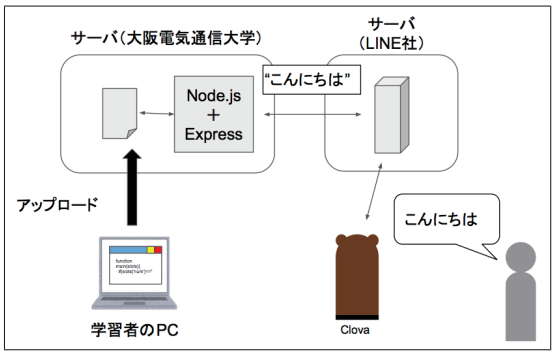


図 5　提案システムの構成図
Fig. 5　Overview of the system.

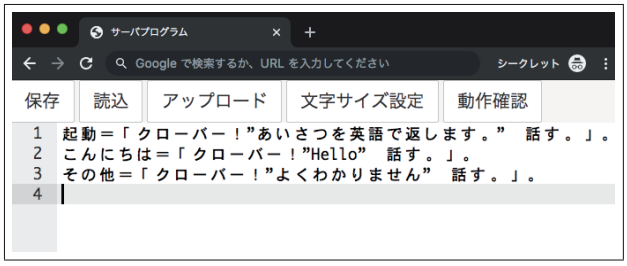


図 6　プログラムの編集画面
Fig. 6　Program editor.

図 7　プログラムの動作シミュレーション画面
Fig. 7　Program simulator.

```
天気＝「クローバー！”今日は晴れです”話す。」。
```

図 8　「天気」という入力語に応答するプログラム例
Fig. 8　A program that responds to the word “weather”.

4.2　プログラムの編集と実行

図 6 に，Web ブラウザ上で動作するプログラムの編集画面を示す．行番号のある部分はテキストエディタになっており，プログラムを入力する．プログラムの保存や読込，動作確認や実機へのアップロードもエディタ上部のボタンを押すことで行う．

プログラムの実行は，Web ブラウザ上で実行する方法と実機で実行する方法の 2 種類がある．Web ブラウザ上で実行する場合は，エディタ上部の「動作確認」ボタンを押す．Web ブラウザ上にチャット風の画面（図 7）が現れ，スマートスピーカーに呼びかける言葉をキーボードから入力し，送信ボタンを押すことで入力に対する動きを確認することができる．実機で実行する場合は，「アップロード」ボタンを押す．続いて実機に対応する ID を入力することで，作成したアプリがサーバにアップロードされる．これにより，特定のスマートスピーカーからアプリを呼び出して使用することができるようになる．

4.3　プログラミング言語の対応

プログラムを記述する言語はドリトル [22] に対応している．ドリトルは，教育用に開発されたプログラミング言語であり，高等学校で多くの実践事例 [7], [8] がある．

Clova のプログラムは，入力語に対応した処理を関数で定義する形で作成する．図 8 に，人が「天気」と話しかけると，Clova が「今日は晴れです」と話すプログラム例を示す．「＝」の左辺は関数名であり，右辺の「」のブロック内に関数の動作を定義する．

この例のように，単純なプログラムについては 1 行で記述することができる．プログラムを簡潔に記述できるようにするために，次の点を工夫した．まず，入力語と対応する動作を，関数名と関数の本体に記述できるようにした．実行時は，入力語に対応する関数が自動的に実行されるようにした．入力語に応じた分岐処理を記述する必要がなくなったことで，プログラムを大幅に簡略化して記述できるようになった．

機能は継続して開発を進めている．入力語に対応した関数の実行は，当初は単一の語に限定していたが，執筆時点では形態素解析を行うことで「大阪の天気」のような複数の語を含まれる入力に対応している．また，入力語の履歴を検索することで，「大阪」という入力語の後の会話で「天気」が入力されたときの動作を定義できるようにした．

4.4　Web API を利用したデータの取得

プログラム開発支援環境では，インターネットのサーバと通信するためのドリトルの Web クライアントオブジェクトを利用して，Web API を使用したプログラムを作成できる．図 9 に，OpenWeatherMap [23] が提供している API を利用し，全世界の気象データを取得するプログラム例を示す．大阪府の気象データを取得するには，指定する URL 中に「`q=Osaka,jp`」のように検索条件を記述する．このように URL と GET リクエストをプログラム中に記述することで，プログラミング言語に用意されたライブラ

```
api1=”https://api.openweathermap.org/data/2.5/weather?q=”。
api2=”,jp&units=metric&appid=XXX ”。
大阪＝「
  天気データ＝Web クライアント！（api1 ＋”Osaka”＋ api2）作る。
  クローバー！（天気データ！気温？）話す。
」。
```

図 9　大阪府の気象データを取得するプログラム例
Fig. 9　A program which gets weather data of Osaka.


```
{
  "coord":{"lon":135.5,"lat":34.7},
  "weather":[{
    "id":520, "main":"Rain",
    "description":"light intensity shower rain",
    "icon":"09d"
   }],
  "base":"stations",
  "main":{
    "temp":17.67, "pressure":1007, "humidity":77,
    "temp_min":13.33, "temp_max":20
   },
  "visibility":10000,
  "wind":{"speed":4.1, "deg":310},
  "clouds":{"all":75},
  "dt":1573457720,
  "sys":{
    "type":1, "id":8032, "country":"JP",
    "sunrise":1573421237, "sunset":1573459002
   },
  "timezone":32400,
  "id":1853909,
  "name":"Osaka",
  "cod":200
}
```


図 10　Web API で取得した JSON 形式データの例
Fig. 10　A JSON format data returned by Web API.

表 1　天気データの取得に用意した命令の例
Table 1　Commands to get weather data.

命令	動作
気温？	取得したデータから気温を読み込む．
最高気温？	取得したデータから最高気温を読み込む．
最低気温？	取得したデータから最低気温を読み込む．

リ関数だけでなく，インターネットで公開されている API をプログラム中から呼び出して利用することが可能であることを意識させることができると考えた．

OpenWeatherMap では，観測地の座標，天気，気温，気圧，風速，などの気象データを取得できる．JSON 形式で取得したデータ例を**図 10** に示す．利用頻度が高い気温系のデータについては，生徒が利用しやすいように**表 1** の命令を用意した．「天気データ！気温？」のように記述することで簡潔に気温を取得できる．気温以外のデータを取得する場合は「風速＝天気データ：`wind:speed`」のように記述することで任意のデータを取得できるようにした．この例を図 10 の JSON 形式のデータに適用すると「`"wind":{"speed":4.1,"deg":310}`」の部分から風速のデータを取得し，「4.1」という値を返す．

表 2　授業計画
Table 2　Lesson plan.

時限	実践内容
1	入力語に対して決まった応答を返すプログラム
2	複数の入力語ごとに応答を返すプログラム
3	ランダムな応答を返すプログラム
4	Web API を利用して気温を返すプログラム
5	Web API で気温以外の情報を返すプログラム
6	Web API で幅広い地域の気象情報を返すプログラム
7	振り返りと授業アンケートなど

5.　高等学校での実践

5.1　実践の概要

スマートスピーカーのプログラム開発支援環境を利用し，高校生がスマートスピーカー専用のアプリを開発することができることを確認するために高等学校の授業で利用した．学習者の前提知識として，基本的なプログラムの動きや Web ページの閲覧時にサーバとやりとりが行われていることは情報の授業で学習済みである．

対象は総合学科の高校 3 年生 14 名である．授業を行った科目は選択科目であり，生徒はドリトルによるプログラミングを体験している．授業はスマートスピーカーのアプリ開発を体験することを目的とした．体験をとおしてスマートスピーカーのアプリを自ら開発できることや，音声認識やアプリの実行はネットワークを介して行っているといったスマートスピーカーの仕組み，Web 上で公開されているデータを使用することができる Web API がある，といったことについて知ることができると考えた．授業計画を**表 2** に示す．

スマートスピーカーは Clova を 1 台用意した．教員が指定した生徒が実機への転送を行い，アプリの動作を確認した．

5.2　スマートスピーカーが話すプログラム（第 1，2 時）

第 1，2 時限目の授業では，スマートスピーカーの基本的な動きを理解するために，人の呼びかけに応じてスマートスピーカーが話すプログラムを作成した．

第 1 時限目は使用する開発環境の使用方法を説明後，アプリ起動時にスマートスピーカーが話す言葉や特定の言葉を呼びかけたときに話すプログラムを作成した．音声認識やアプリの実行がサーバ上で行われていることについてはスライドを見せながら説明した．

第 2 時限目は，前時限の内容をふまえて血液型診断をするアプリや夕食を提案するアプリを作成した．夕食を提案するアプリのプログラム例を**図 11** に示す．このプログラムは，「お腹すいた」，「和食」，「洋食」，「中華」の単語をスマートスピーカーに呼びかけると，それぞれの単語に対応した言葉を話す．それ以外の言葉を話しかけると，「夕食

```
お腹すいた＝「
  クローバー！”和食，洋食，中華のどれを食べたい？”話す。
」。
和食＝「クローバー！”夕食は肉じゃがです”話す」。
洋食＝「クローバー！”夕食はオムライスです”話す」。
中華＝「クローバー！”夕食は餃子です”話す」。
その他＝「クローバー！”夕食はトムヤムクンです”話す」。
```

図 11　夕食を提案するアプリのプログラム例
Fig. 11　A program which speaks dinner recommendation.

```
お腹すいた＝「
  クローバー！”和食、洋食、中華のどれを食べたい？”話す。
」。
和食＝「
  数＝乱数（3）。
  「数＝＝ 1」！なら「クローバー！”夕食は肉じゃがです”話す」実行。
  「数＝＝ 2」！なら「クローバー！”夕食は筑前煮です”話す」実行。
  「数＝＝ 3」！なら「クローバー！”夕食はサバ味噌です”話す」実行。
」。
```

図 12　ランダムな結果を返すプログラム例
Fig. 12　A program which speaks random messages.

は，トムヤムクンです」と話す．たとえば，アプリを起動し，「和食」と呼びかけると，スマートスピーカーが「夕食は，肉じゃがです」と話す．

5.3　乱数を利用したプログラム（第 3 時）

第 1，2 限目で扱ったプログラムは，同じ単語を何度呼びかけてもスマートスピーカーは同じ言葉を話すものであった．そこで第 3 時限目は同じ単語でも呼びかけるごとにスマートスピーカーが話す言葉を変えられることを体験するために，乱数の値を利用したプログラムを作成した．まず，あらかじめ用意しておいたサンプルプログラムを生徒全員が入力・実行し，プログラムがどのような動きをするのかを確認した．その後，前時限で作成した夕食を提案するアプリ（図 11）を乱数の値によって提案する料理を変更できる形に改良した．生徒が改良したプログラムの一部を**図 12** に示す．

5.4　Web API を利用したプログラム（第 4，5，6 時）

第 4，5，6 時限目は，企業などが公開・提供しているデータをプログラムで利用する技術があることを体験するために，Web API を利用したプログラムを作成した．今回はスマートスピーカーで利用されており，生徒に身近な気象データを取得する Web API を利用した．

第 4 時限目は，Web API の概要について説明後，生徒全員で図 9 の大阪府の気温を取得するサンプルプログラムを入力・実行し，プログラムの動きを確認した．動きを確認後，プログラム中に記載されている Web API の URL を Web ブラウザでアクセスし，データがどのような形式で取得できるのか，どのようなデータが取得できるかを確認した．その後，生徒は大阪府以外の都道府県の気温を取得する課題に取り組み，気温に応じてスマートスピーカーが話す言葉を変えるアプリを作成した．プログラム例を**図 13** に示す．このプログラムは，「大阪」と呼びかけると，大阪府の気温を取得し，取得した気温が 25 度より低いときにはスマートスピーカーが「今晩はチゲ鍋はいかがでしょう」と話し，そうでないときは「今晩は素麺はいかがでしょう」と話す．

```
api1=”https://api.openweathermap.org/data/2.5/weather?q=”。
api2=” ,jp&units=metric&appid=XXX ”。

大阪＝「
  天気データ＝ Web クライアント！（api1 ＋” Osaka”＋ api2）作る。
  「（天気データ！気温？）＜ 25」！なら「
    クローバー！”今晩はチゲ鍋はいかがでしょう”話す。
  」そうでなければ「
    クローバー！”今晩は素麺はいかがでしょう”話す。
  」実行。
」。
```

図 13　取得した気温によって処理を変更するプログラム
Fig. 13　A program which speaks messages corresponding to temperature.

```
api1=” https://api.openweathermap.org/data/2.5/weather?q=”。
api2=” ,jp&units=metric&appid=XXX ”。

大阪＝「
  天気データ＝ Web クライアント！（api1 ＋” Osaka”＋ api2）作る。
  クローバー！（”大阪の風速は”+天気データ：データ：wind：speed
+”です”）話す。
」。
```

図 14　大阪の風速を取得するプログラム
Fig. 14　A program which gets wind speed of Osaka.

第 5 時限目は，Web API で取得できる気象データを復習後，気温以外のデータを取得するプログラムを作成した（**図 14**）．加えて，前時限までは取得した気温を話すことしかできなかったが，今回は取得したデータと文字列を組み合わせるプログラムも扱った．生徒は気圧や天気を取得する課題に取り組んだ後，乱数を組み合わせてスマートスピーカーが天気に応じたおすすめスポットをいうアプリを作成した．

第 6 時限目は，前時限の内容を復習した後に対話的なプログラムを作成した．プログラム例を**図 15** に示す．これは，アプリ起動時に「気象情報を知りたい地名を教えてください」とスマートスピーカーが話し，「札幌」か「沖縄」と呼びかけると，呼びかけられた場所の気温を取得し，スマートスピーカーが話すプログラムである．生徒は図 15 のサンプルプログラムを入力し，動作を確認後，課題として他の都道府県や海外の都市の気象データを取得するプログラムを作成した．

5.5　授業アンケート（第 7 時）

第 7 時限目は，生徒に対してアンケートを実施した．質

```
api1=”https://api.openweathermap.org/data/2.5/weather?q=”。
api2=”,jp&units=metric&appid=XXX ”。

起動＝「クローバー！”気象情報の地名を教えてください”話す」。
札幌＝「場所＝”札幌”。クローバー！”何を知りたいですか”話す。」。
沖縄＝「場所＝”沖縄”。クローバー！”何を知りたいですか”話す。」。

気温＝「
  「場所＝＝”札幌”」！なら「
    天気＝Web クライアント！（api1 ＋”Sapporo”＋ api2）”作る。
    クローバー！（”札幌の気温は”＋（天気！気温？）＋”です”）話す。
  」実行。
  「場所＝＝”沖縄”」！なら「
    天気＝Web クライアント！”（api1 ＋”Okinawa”＋ api2）作る。
    クローバー！（”沖縄の気温は”＋（天気！気温？）＋”です”）話す。
  」実行。
」。
```

図 15　対話的に天気を話すプログラム例
Fig. 15　A program which speaks weather forecast interactively.

Q1. 授業は楽しかった
Q2. 授業内容について関心をもった
Q3. このような授業を今後も受けてみたいと思った
Q4. 授業の学習内容について理解できた
Q5. 授業の学習内容は難しかった
Q6. 授業を受けて、新しいことを学ぶことができた
Q7. 今後の授業で、Clova を使ったどんなアプリ（スキル）を作ってみたいですか。「～を作ってみたい」「～を作れるようにしてほしい」など具体的に書いてください。
Q8. 今回作成した天気のように、ネットで提供されているデータやサービスを利用しているスマホやパソコンのアプリには何があるでしょうか。（複数回答可）
Q9. 感想や意見、気づいたことなど、何かありましたら自由に書き込んでください（自由記述）。

図 16　アンケートの質問項目
Fig. 16　Questinnaires for students.

問項目を**図 16** に示す．Q1～Q6 は，授業への興味・関心を問う質問である．生徒は 1～5 の 5 段階で回答する．質問に対してまったくあてはまらない場合は「1」，あてはまる場合は「5」の数字を選択する．Q7～Q8 は，記述式で回答する．Q7 は作成したいスマートスピーカーのアプリについて問う質問である．スマートスピーカーでできることと，できないことを理解していれば，実現できるアプリや使用されている技術を適切に回答することができると考えた．Q8 は Web API について問う質問である．Web API が企業などが提供しているデータをプログラムで利用するためのものであることを理解していれば，Web API を利用したアプリを想定し回答することができると考えた．生徒には口頭で思いつくだけ記述するように伝えた．Q9 は，今回の授業についての感想などを自由記述で回答する．

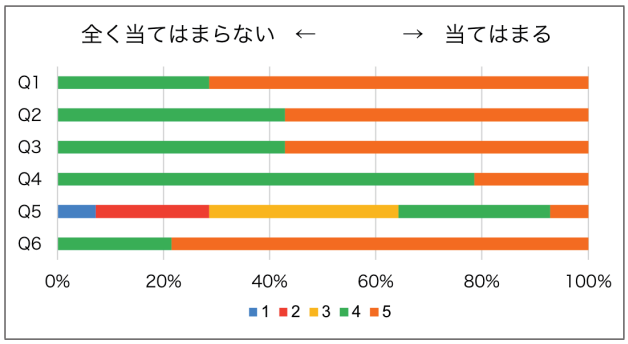


図 17　アンケート結果（Q1～Q6，N=14）
Fig. 17　Result graphs for Q1–Q6.

表 3　質問ごとの回答数（Q1～Q6，N=14）
Table 3　Results for Q1–Q6.

質問	1	2	3	4	5
Q1	0	0	0	4	10
Q2	0	0	0	6	8
Q3	0	0	0	6	8
Q4	0	0	0	11	3
Q5	1	3	5	4	1
Q6	0	0	0	3	11

6.　実践した結果

6.1　授業内容に関する評価（Q1～Q6，Q9）

生徒に対して行った Q1～Q6 のアンケート結果を**図 17** と**表 3** に示す．1～5 の数のうち選択した数字が大きいほど，「あてはまる」と生徒は回答している．

Q1～Q4 と Q6 では，1～3 を選択した生徒はいなかった．Q1，Q2，Q3，Q6 は 5（あてはまる）を選択した生徒が半数以上を占めていることから，今回の授業は生徒にとって新しい内容で楽しく関心を持って取り組める内容であったことが分かる．

授業の内容について理解できたかを問う Q4 では 4 を選択した生徒が 11 名（42%），5 を選択した生徒が 3 名（21%）だったことから，内容についても理解することができたと生徒が思えたことが分かる．

授業の難易度を問う Q5 では，それぞれ 1 が 1 名（7%），2 が 3 名（21%），3 が 5 名（35%），4 が 4 名（28%），5 が 1 名（7%）選択していたことから，生徒にとって適切な難易度であったことが分かる．

Q9 に対する回答を**図 18** に示す．Q9 は自由記述とし，授業の感想や意見などを問う質問である．生徒の感想から，スマートスピーカーの裏側で動作する音声認識の AI や Web API など新しい技術を学ぶことができた授業になっていることが分かった．

6.2　スマートスピーカーに関する理解度（Q7）

Q7 は今後どのようなスマートスピーカーのアプリを作っ

最近になってよくスマートスピーカーという言葉を耳にすることが増えましたが、そのアプリを実際に作ることができるのに感動しました。めったに触れる機会のないことなので、それを授業でできて、とても良い経験になりました。
天気がどのように調べられて自分に届くのか詳しく知ることができてよかった
自分の知らないことや知らない技術を学ぶことができてとても有意義で楽しかった
自分好みにカスタマイズするのが楽しかった。ただ、もっと作り変える時間が欲しかった。

図 18　生徒の感想（Q9，自由記述）
Fig. 18　Student reflections.

表 4　生徒が作成したいアプリ（Q7，N=14）
Table 4　Applications that students want to develop.

生徒の回答	回答数
音楽を再生するアプリ	4
時間割を確認するアプリ	3
しりとり	1
スポーツ結果の速報	1
服装提案	1
友達からのメッセージを知らせる	1
本の朗読	1
質問に答えることで旅行先を提案するアプリ	1
都道府県ごとの方言を喋るアプリ	1

てみたいかを問う質問である．スマートスピーカーができることを理解していれば，実現可能なアプリを回答することができると考えた．生徒の回答を**表 4** に示す．いずれの回答もスマートスピーカーで実現可能なアプリを回答していることから，生徒はスマートスピーカーでできることと，できないことを理解していることが分かる．

6.3　Web API に関する理解度（Q8）

Q8 は Web API が利用されているサービスについて問う質問である．Web API がどのような場面で利用されるものかを理解していれば，利用されているサービスを想定し回答することができると考えた．回答を**表 5** に示す．

13 件の回答を見ると，PC やスマートフォンなどのアプリケーションからインターネットのサービスを利用する可能性のあるサービスを回答していることから，生徒は Web API を利用したアプリ開発をとおして，Web API が Web 上で提供されているデータをプログラムで利用する仕組みがあることを学ぶことができたことが分かる．

6.4　教員への聞き取り調査による評価

授業を担当した教員 2 名に対して，インタビュー調査を行った．回答を**図 19** に示す．インタビュー調査は第 3 時と第 7 時の終了後に，口頭で授業について聞き取りを行った．

インタビュー調査から，生徒は面白そうに実習に取り組んでおり，スマートスピーカーを活用した授業について教員は好感をもっていることが分かった．生徒が記述するプログラムの難易度については，ドリトル言語で記述できることによりちょうどよい難易度であると感じていることが分かった．また，世界の気象データを取得できる Web API を利用したプログラムについては，気候区分など地理との教科横断的な学びの可能性を見い出すことができた．ま

表 5　Web API が利用されていると思うサービス（Q8，複数回答）
Table 5　Applications in which students thought Web API is used.

生徒の回答	回答数
地図（Google Map も含む）	4
電車の乗り換え，運行情報	2
株価	1
占い	1
世界各国の標準時間	1
カーナビ	1
Amazon	1
Yahoo!	1
Safari	1

生徒の様子について
- 生徒は面白そうに取り組んでいた．
- 生徒は Clova に話させる言葉を何にしようかと考えていた．

授業の展開について
- 第 1，2 時限目の授業で「話すだけだと面白くない」ということに気づかせることが、発展していく上で重要になる．
- 第 3 時限目の授業で Clova に話させる言葉は，生徒がすぐに思いつく題材にする必要があると感じた．乱数を利用するにあたって多くの言葉を扱う必要がある場合，「料理」は調べることができるので適しているが、「天気」や「あいさつ」は選択肢が少ないため扱うのは難しい．
- Clova が話すだけでなく，もう少し展開が広がるとよいと感じた．その点で Web API を利用することで色々と幅が広がるのでよかった．
- Web API によって世界の国々の気象データを取得できるため，気候区分など地理との教科横断的な授業が可能になるかもしれない．

授業でスマートスピーカーを活用することについて
- JavaScript は難しかったので，システムがドリトルに対応していてよかった．
- 今回の授業を情報 I で実施する場合は，時数的に難しいかもしれないため，情報 II のほうが活用しやすいと感じた．
- 今の高校生にはちょうどよい難易度であるが，数年後の高校生には簡単すぎるかもしれない．
- 高校では実用的な技術に寄っていきたいため，Clova を単独で利用するのではなく，API や機器，ブラウザ，スマートフォンと連携するような教材にするとよい
- 実際にあるものの仕組みを見せるという意味で，家電の制御ができればよい．

図 19　教員のコメント
Fig. 19　Comments of teachers.

た，スマートスピーカーを単独で活用するだけではなく，他の製品やサービスと連携した教材を求めていることが分かった．

7. 考察

第 7 時限目に実施した生徒へのアンケート結果と教員へのインタビュー調査から，スマートスピーカーのアプリ開発を題材にした授業について考察する．

生徒へのアンケート結果から，スマートスピーカーのアプリ開発を題材にした授業は楽しく関心を持って取り組める内容であったことを確認した．また，作成したいスマートスピーカーのアプリを問う質問についてほとんどの生徒が適切な回答をしていたことから，スマートスピーカーのアプリ開発をとおして，スマートスピーカーができることを学ぶことができたといえる．スマートスピーカーは音声で何かを伝えると返答する仕組みだと考える生徒もいた．これは，使用したスマートスピーカーの台数の関係上，プログラムの確認を画面上で実行することが多かったため，スマートスピーカーがネットワーク通信を行い，アプリを実行していることを意識させにくかったことが理由の 1 つとして考えられる．そのため，作成したアプリをスマートスピーカーで複数回，実行する必要があると考えられる．

Web API について問う質問についてもほとんどの生徒が適切な回答をしていた．Web API としてデータを提供している企業名や機器をあげた回答も，Web API が企業などが Web 上で公開・提供しているデータをプログラムで利用する技術であることは理解していると考えられる．また，自由記述による感想（図 18）で「天気がどのように調べられて自分に届くのか詳しく知ることができてよかった」，との記述があることから，生徒は Web API について体験的に学ぶことができたことが読み取れる．

教員へのインタビュー調査から，生徒が面白そうに実習に取り組んでいることを確認した．インタビューではスマートスピーカーを活用した授業について様々な提案があったことから，教員がスマートスピーカーのアプリ開発を題材にした授業について，好意的な印象を持っていることが読み取れる．また，「JavaScript では難しかったので，システムがドリトルに対応していてよかった」との発言があったことから，開発したプログラム開発支援環境を利用する事で高校生がスマートスピーカーのアプリを開発できたことを確認した．

8. おわりに

本論文では，スマートスピーカーのアプリ開発に特化したプログラム開発支援環境を開発した．Web ブラウザ上で動作し，教育用プログラミング言語に対応することで，学校現場に導入する際の課題を解決できた．

高等学校で授業を行い，生徒へのアンケート結果と教員へのインタビューにより，高校生がスマートスピーカー専用のアプリを開発できることを確認した．また，Web API を利用したスマートスピーカーのアプリ開発をとおして，Web API が Web 上で提供されているデータをプログラムで利用する仕組みであり，どのような場面で利用されているのかを生徒が学ぶことができたことを確認した．

謝辞　本研究は，科学研究費補助金（基盤研究（C）17K00989）の補助を受けています．兵庫県立西宮今津高等学校の白井美弥子先生と吉成良子先生には，今回の実践について多大なご協力をいただきました．感謝いたします．

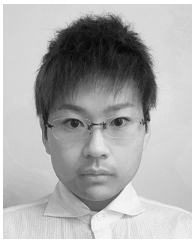
参考文献

[1] 文部科学省：高等学校学習指導要領解説 (2017).
[2] 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，pp.561–562 (2018).
[3] Tagawa, T., Jin, M. and Inoue, H.: A Smart Speaker Application to Assist Japanese Onomatopoeia Learning: A Prototype, *Proc. Society for Information Technology & Teacher Education International Conference 2019*, pp.1124–1127 (2019).
[4] 吉田裕行，中村晃輔，松尾洋幸，日熊隆則，岡本牧子，宮田龍太：算数文章問題の学習を支援するスマートスピーカーのスキル開発，研究報告コンピュータと教育（CE），Vol.2018-CE-147, No.2, pp.1–6 (2018).
[5] 平尾康起，和田　健，前田篤志：メカトロニクスコースにおけるプログラミング実験実習，大阪府立大学工業高等専門学校研究紀要，Vol.52, pp.47–50 (2018).
[6] 鶴見昌代，宮城愛美：視覚障害者によるスマートスピーカー活用の可能性（サイトワールド 2018 出展報告），筑波技術大学テクノレポート，Vol.26, No.2, pp.74–79 (2019).
[7] 間辺広樹，大村基将，林　康平，兼宗　進：情報科教育における IoT 学習環境の利用方法の検討，情報処理学会，情報教育シンポジウム（SSS2016），pp.98–105 (2016).
[8] 間辺広樹，長島和平，並木美太郎，長　慎也，兼宗　進：高等学校における複数言語によるプログラミング教育の提案，情報処理学会論文誌「教育とコンピュータ」，Vol.3, No.3, pp.29–41 (2017).
[9] 兼宗　進，白井詩沙香，竹中一平，長瀧寛之，小林史弥，島袋舞子，田邉則彦：データベースと情報システムを学習する授業の提案と実践，情報処理学会論文誌「教育とコンピュータ」，Vol.3, No.3, pp.18–28 (2017).
[10] LINE: Clova Developer Center, available from ⟨https://clova-developers.line.biz/⟩ (accessed 2020-02-02).
[11] Amazon: Amazon 開発者ポータル，入手先 ⟨https://developer.amazon.com/ja/⟩（参照 2020-02-02）.
[12] Google: Google Developers, available from ⟨https://developers.google.com/⟩ (accessed 2020-02-02).
[13] Amazon: Alexa Skill Blueprints, available from ⟨https://blueprints.amazon.co.jp/⟩ (accessed 2020-07-04).
[14] cocorita, available from ⟨https://cocorita.jp/⟩ (accessed 2020-07-05).
[15] 株式会社アイリッジ：NOID，入手先 ⟨https://www.noid.ai/⟩（参照 2019-09-18）.
[16] Voiceflow, available from ⟨https://www.voiceflow.com/⟩ (accessed 2019-09-18).
[17] VoiceApps, available from ⟨https://voiceapps.com/⟩ (accessed 2020-07-05).
[18] SmartHouse Technologies: BotTalk, available from

⟨https://bottalk.de/⟩ (accessed 2019-09-18).
[19] 本多佑希，島袋舞子，兼宗　進：スマートスピーカーを題材にしたプログラミング学習環境の提案，研究報告コンピュータと教育（CE），Vol.2019-CE-150, No.5, pp.1–5 (2019).
[20] 本多佑希，島袋舞子，浅子秀樹，兼宗　進：スマートスピーカーのアプリケーション開発を支援するプログラミング学習環境の開発，情報教育シンポジウム論文集，Vol.2019, pp.62–68 (2019).
[21] Inho, K.: Clova: Services and Devices Powered by AI, *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp.1359–1359 (2018).
[22] 兼宗　進，御手洗理英，中谷多哉子，福井眞吾，久野　靖：学校教育用オブジェクト指向言語「ドリトル」の設計と実装，情報処理学会論文誌，Vol.42, No.11, pp.78–90 (2001).
[23] OpenWeatherMap, available from ⟨https://openweathermap.org⟩ (accessed 2019-09-18).

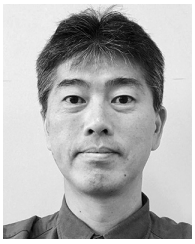
島袋 舞子　（正会員）

2014 年沖縄国際大学産業情報学部産業情報学科卒業．2016 年大阪電気通信大学大学院医療福祉工学研究課程修了．2018 年より大阪電気通信大学大学院工学研究科博士課程に在籍．修士（工学）．2016 年から大阪電気通信大学情報教育特任講師．初等中等教育における情報科学教育に関する研究に従事．日本情報科教育学会会員．



本多 佑希　（学生会員）

2016 年大阪電気通信大学情報学部情報学科卒業．2019 年大阪電気通信大学大学院工学研究科修士課程修了．同年より大阪電気通信大学大学院工学研究科博士課程に在籍．修士（工学）．



兼宗 進　（正会員）

1987 年千葉大学工学部電子工学科卒業．1989 年筑波大学大学院理工学研究科修士課程修了．2004 年筑波大学大学院ビジネス科学研究科博士課程修了．博士（システムズマネジメント）．企業勤務後，2004 年一橋大学総合情報処理センター准教授．2009 年から大阪電気通信大学医療福祉工学部/総合情報学部を経て工学部電子機械工学科教授．プログラミング言語，情報科学教育に興味を持つ．ACM，IEEE Computer Society 各会員．