

予習・復習のための動画コンテンツ提供に関する研究

岩本 康平¹ 高野 辰之² 小濱 隆司³ 宮川 治³

概要: 近年の ICT 技術の発展により動画コンテンツを閲覧して予習・復習を行う方法が普及しつつある。この方法は手軽に利用者のペースで学べるが、動画コンテンツを作成するハードルが高くその導入は容易ではない。また単に動画コンテンツを作成し提供するだけでは学生は適切な難易度のコンテンツで学べず、モチベーションが下がる可能性がある。そこで本研究では学生に合った予習と復習用の動画コンテンツを提供するためのフレームワークを開発する。動画コンテンツは Web ブラウザ上で閲覧する。学生に合った動画コンテンツを提供するために、コンテンツの種別情報などを電子化した教材に含ませた。評価としてプログラミング科目を例に、難易度別にコンテンツを振り分け提供できることを確認した。また動作実験を行い、動画コンテンツが再生できることを確認した。

Study on Provision of Video Content for Preparation and Review

1. はじめに

近年の ICT (Information and Communication Technology) 技術の発展により、教科書やドリルを通じた学修の他に、Web ブラウザやアプリで動画コンテンツを見て学修する方法が普及しつつある。この方法のメリットは場所を選ばず手軽に利用者のペースで学べることである。一方デメリットとしてはコンテンツ作成の負荷や作成コストが大きいことである。e ラーニング戦略研究所の調査 [1] では、教育関係者の多くが動画コンテンツを利用した教育を「わかりやすく学習効果が高い」、「学生・生徒の学習意欲が上がる」と認識し、その活用にきわめて前向きであるとしている。また動画コンテンツの活用方法の調査では講義・授業における補足資料としての活用が最も多く、オンデマンド教材として予習・復習・補講などに活用が最も低かった。

文部科学省は大学教育部会の審議において、予習・復習などを含めた学修時間の増加・確保による主体的な学びの確立に取り組む必要があるとしている [2]。しかし、国立教育政策研究所の大学生等の学習状況に関する調査では、授

業への出席時間に対して予習・復習にかかる時間は十分とはいえないと指摘している [3]。学修する際に、教科書は見て学修するが、動画コンテンツは見て聴きながら学修できるため、学修を始めるハードルが低いと考えられる。現状では予習・復習への動画コンテンツの活用が低いが、予習・復習用の動画コンテンツを提供し学修を始めるハードルを下げること、日々の学修時間の増加が期待できる。

動画コンテンツの作成方法には教員自らが作成する、外部に制作を委託するなどがある。ICT 利活用教育に用いるコンテンツの作成方法に関する調査では、「教員が独力で作成している」という回答が最も多い結果となった。また ICT 利活用教育の推進を阻害する要因の一つとしてシステムやコンテンツを作成、維持する人員の不足が述べられている [4]。このことから、コンテンツを作成する人員が不足しているために、教員が独力で作成する状況に陥っていると考えられる。本来教員は授業の設計や運営に集中すべきであり、コンテンツ作成などの作業に時間を割かれるべきではない。特に動画コンテンツの作成は撮影から編集までの作業が必要となり、講義資料を作成するよりも時間や労力がかかるためハードルは高い。そのため動画コンテンツをより簡単に作成することができればそのハードルは下がると考えられる。

大学の授業は 15 週にわたり実施され、授業によっては教科書のすべての範囲を扱う場合もある。授業内容をもとに予習・復習用の動画コンテンツを作成するとコンテンツ

¹ 東京電機大学大学院情報環境学研究所
Graduate School of Information Environment, Tokyo Denki University

² 関東学院大学理工学部
College of Science and Engineering, Kanto Gakuin University

³ 東京電機大学情報環境学部
School of Information Environment, Tokyo Denki University

の数は授業の範囲に応じて増えていく。しかしコンテンツの数が多くなるとどのようなコンテンツがあるのか、内容はどのようなものを把握するのは難しくなる。また学生一人一人が同じ授業で同じ内容を学んだとしても、学生全員が同じ理解度に達することはない。これは学生がこれまで学んできた素養が異なっていることやそれぞれ興味関心のある事柄が共通ではないからである。その授業に対する理解度の低い学生にとっては用意されているコンテンツをすべて学ぶことが理解度向上への近道となる。一方でその授業に対する理解度の高い学生にとってはその学生に与えるべき難易度のコンテンツよりも難易度の低いコンテンツが与えられ、モチベーションが下がる可能性がある。そのため、単に動画コンテンツを提供するだけでなく、学生に合った動画コンテンツを提供することが重要だと考えられる。

そこで本研究では学生に合った予習と復習用の動画コンテンツを提供するためのフレームワークを開発する。このフレームワークをコンテンツ作成者が利用することで動画コンテンツを容易に作成し、学生に合った動画コンテンツの提供を行える。

2. 関連研究

近年の急速な ICT 発展に伴い、動画コンテンツ生成や配信に関する研究が盛んに行われている。

林ら [5] はテレビ番組をまるまる 1 本記述できる言語 TVML (TV program Making Language) の提案・開発を行っている。これは TVML 言語で作成したい番組内容を記述するだけで自動で動画コンテンツを生成できる。本研究で提供する動画コンテンツは予習や復習などの学修に用いることを想定しているため、林らの開発している TVML とは異なる。

市村ら [6] は講義を撮影、配信までを自動で行う講義自動収録システムを開発している。このシステムはビデオカメラを 1 台設置するだけで黒板を用いた講義の動画を容易に作成し、即座に配信することができる。市村らは黒板を用いた講義を e ラーニング教材にし即座に配信することを目的としているが、本研究では単に動画コンテンツを提供するだけに留まらず、学生に合った動画コンテンツの提供を目的としたフレームワークを開発している点で異なる。

八重樫ら [7] は対面講義を自動収録し講義コンテンツを自動生成するシステムを開発している。このシステムは対面講義で使用した板書、書画カメラ、電子スライドを組み合わせて講義コンテンツを作成する。本研究では対面講義の撮影は一切行わず、すでに存在する講義資料の内容をもとに動画コンテンツを動的に生成・提供するため、八重樫らの開発したシステムとは異なる。

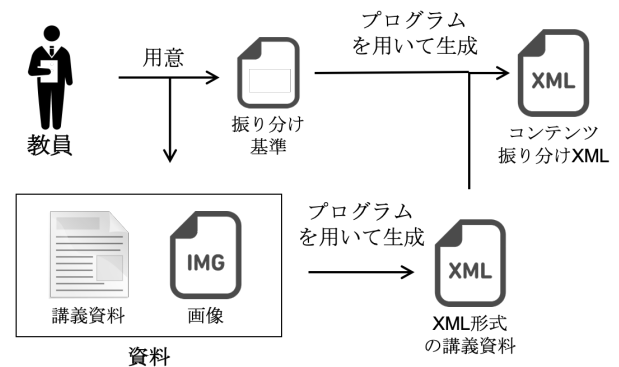


図 1 動画コンテンツ提供の下準備の流れ

Fig. 1 Flow of preparation for providing video content

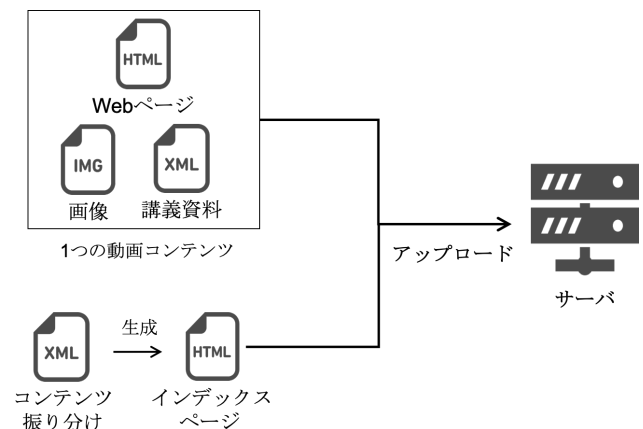


図 2 動画コンテンツ提供の流れ

Fig. 2 Flow of provision of video content

3. システム概要

本研究では学生に合った予習と復習用の動画コンテンツを提供するためのフレームワークを開発する。これを活用することで容易に動画コンテンツの提供を行える。

本フレームワークでは、動画コンテンツの生成機能と学生に合った動画コンテンツ群生成に必要な要素を提供する。本研究で述べる動画コンテンツ群とは、生成した動画コンテンツのインデックスである。動画コンテンツの生成は、動画コンテンツのもととなる電子化した教材の内容を用いる。この電子化した教材を本研究では、XML 形式の講義資料と呼ぶ。動画コンテンツは Web ブラウザ上で動作し、学生が Web ページにアクセスすることでプログラムが動作し動画コンテンツが動的に生成される。動画コンテンツを動的に生成することで容量の大きい動画ファイルをサーバに置かずに済むため、動画コンテンツの導入が容易になることが期待できる。また動画コンテンツは Web ページに自由に組み込むことができる。

学生に合った動画コンテンツを提供するには、学生自身の成績や学修時間などの情報をもとにすると考えられる。しかし講義を担当する教員により成績を基準にするのか、

あるいは学修時間を基準にするのが異なる場合がある。本研究では成績や学修時間などの具体的な振り分け基準をもとにした学生に合ったコンテンツ群を構成すること自体には注力せず、動画コンテンツ群を構成するために必要となる情報を XML 形式の講義資料に関連させた。動画コンテンツ群を構成するための情報を XML 形式の講義資料に含ませることで、動画コンテンツ群を構成するための振り分け基準を教員が自由に決めることができる。

次に教員が本フレームワークを利用して動画コンテンツを提供するまでの流れを述べる。図 1 に動画コンテンツ提供の下準備の流れを示す。教員は提供する動画コンテンツのもととなる講義資料とその講義資料で使われる画像を用意する。講義資料は動画コンテンツに必要な要素だけを抜き出し XML 形式の講義資料を生成する。その後、その講義資料がどのようなコンテンツかという情報などを XML 形式の講義資料に追加する。この情報は動画コンテンツ群を生成する時に用いる。追加した情報をもとに、教員が設定した振り分け基準に沿って動画コンテンツ群の情報をコンテンツ振り分け XML として生成する。

図 2 に動画コンテンツ提供の流れを示す。動画コンテンツ提供の下準備が完了したら、Web ページを作成する。この Web ページには 1 つの動画コンテンツが組み込まれる。Web ページの中で画像と XML 形式の講義資料を指定し動画コンテンツの生成機能を利用するための簡単なスクリプトを記述する。最後に Web ページ、XML 形式の講義資料、画像ファイルをサーバにアップロードする。これで 1 つの動画コンテンツが提供できる状態となる。複数の動画コンテンツを提供する場合には同じ手順で作成しサーバにアップロードし、コンテンツ振り分け XML をもとに動画コンテンツのインデックスページを作成しサーバにアップロードする。

次に学生が利用する流れを説明する。学生はサーバにアップロードされているインデックスページにアクセスする。インデックスページには学生に合った動画コンテンツ群が表示され、その中から動画コンテンツを選んで学修する。

4. システム詳細

4.1 XML 形式の講義資料の生成

動画コンテンツを生成するために XML 形式の講義資料が必要となる。XML 形式の講義資料は動画コンテンツ生成に使用する情報と、動画コンテンツ群生成に使用する情報を持つ。

本研究では XHTML 形式の講義資料を XML 形式の講義資料生成プログラムを用いて XML 形式の講義資料を生成する。このプログラムは Java 言語を用いて作成し、DOM (Document Object Model) API を用いて XHTML の情報を解析、JAXB (Java Architecture for XML Binding) を

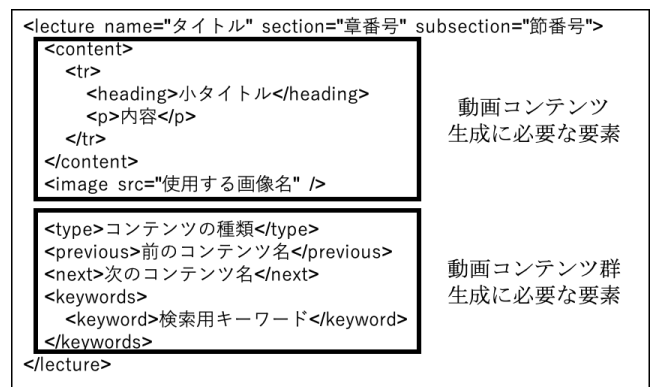


図 3 XML 形式の講義資料

Fig. 3 Lecture materials in XML format

用いて XML を生成する。XHTML 形式の講義資料には講義の内容や Web ページを構成するための情報が含まれている。XHTML 形式のままでは講義資料に直接関係のない情報まで含まれているため、動画コンテンツ生成に必要な要素に絞り込む。XML 形式の講義資料に動画コンテンツ生成に必要な要素だけに絞り込んだことで、再利用や拡張が容易になる。XML 形式の講義資料生成後、学生に合った動画コンテンツ群生成のための要素を XML 形式の講義資料に追加していく。現状では動画コンテンツ生成に必要な講義資料の内容を抜き出し XML 化する部分に関しては自動で行うことができる。しかし学生に合った動画コンテンツ群生成のための要素については現段階では手動で設定する必要がある。これについては XML 形式の講義資料の要素で詳しく述べる。

また本研究ではもとの講義資料を XHTML 形式としているが、情報を解析する部分を入れ替えるだけで他の形式の講義資料にも対応可能である。

4.2 XML 形式の講義資料の要素

ここでは XML 形式の講義資料に必要な要素について説明する。図 3 に XML 形式の講義資料を示す。XML はルートとなる要素を 1 つ持つ必要がある。そのため lecture 要素をルート要素とし、属性に対象となるコンテンツのタイトルを持つ name、章番号を持つ section、節番号を持つ subsection を持たせた。このルート要素の中に動画コンテンツを提供するための 2 種類の情報が入っている。

1 種類目は動画コンテンツ生成に必要な要素である。ここには講義資料の説明文などの情報が入る。次に要素の詳細について説明する。content 要素に講義資料の内容がまとめられ、tr 要素は小タイトルとそれに対応する内容がまとめられる。heading 要素には小タイトルが入り、p 要素には小タイトルの内容が入る。image 要素の src 属性に使用する画像名が入る。

2 種類目は学生に合った動画コンテンツ群生成に必要な要素である。ここには学生に合った動画コンテンツ群を生

```

<index>
  <section number="1">
    <content>A1_1</content>
    <content>A1_2</content>
  <section>
    <section number="2">
      <content>A2_1</content>
      <content>A2_2</content>
      :
    </section>
    :
  </index>

```

図 4 コンテンツ振り分け XML
Fig. 4 Content distribution XML

成するために使用する要素が入る。次に要素の詳細について説明する。type 要素にはそのコンテンツの種類が入る。コンテンツの種類は予習、復習、演習、演習解答の4つを定義した。previous 要素には前のコンテンツ名、next 要素には次のコンテンツ名が入る。keyword 要素にはそのコンテンツの検索用キーワードが入り、keywords 要素が keyword 要素をまとめる。

現状では動画コンテンツ群を生成するために必要となる要素は手動で設定している。講義資料を作成した教員でなければ、対象となるコンテンツがどのような種類のものなのか判別がつかない。設定する要素の数はコンテンツの数に応じて増えるため、将来的には手動ではなく、自動で要素の設定を行っていく必要がある。また学生に合った動画コンテンツ群生成に必要な要素を拡張していくことも可能である。現状では4つの要素を含ませているが、今後実際に教員に使用していただくことでどのような要素が必要になるのかがより具体的になると考えられる。

4.3 コンテンツ振り分け XML

コンテンツ振り分け XML は動画コンテンツ群を作成するために使用する。XML 形式の講義資料の情報と教員が作成した動画コンテンツの振り分け基準の情報をもとに、コンテンツ振り分け XML 生成プログラムを用いてコンテンツ振り分け XML を生成する。

図 4 にコンテンツ振り分け XML を示す。コンテンツ振り分け XML の要素は3つある。1つ目はすべての要素をまとめる index 要素である。2つ目は章単位のコンテンツをまとめる section 要素である。section 要素では number 属性で指定した章のコンテンツをまとめる。3つ目はコンテンツ情報を記した content 要素である。content 要素にはコンテンツの名前が値として入る。



図 5 動画コンテンツの再生画面
Fig. 5 Playback screen of movie content



図 6 再生ボタンと一時停止ボタンの画面
Fig. 6 Playback button and pause button screen

4.4 動画コンテンツ

動画コンテンツはブラウザから Web ページにアクセスして利用する。動画コンテンツは動画コンテンツ再生スクリプトを生成する機能と動画コンテンツを描画する機能がある。

動画コンテンツ再生スクリプトを生成する機能では、JavaScript の DOM API を用いて XML 形式の講義資料を解析し、解析した結果をもとに動画コンテンツ再生スクリプトを生成する。このスクリプトは Web ページにアクセスした段階で動的に生成される。

動画コンテンツを描画する機能では、生成された動画コンテンツ再生スクリプトをもとに動画コンテンツの描画と音声出力を行う。描画には Three.js を使い、音声出力には WebSpeechAPI を用いた。Three.js[8] は JavaScript API を通じて WebGL の機能を非常に簡単に利用でき、美しい 3D グラフィックスが作成できる。Web Speech API とは Web ブラウザ上で音声データを扱うことのできる API である。描画で使用するアバターは 3D モデルの MMD モデル*1を用いて表示する。

MMD (MikuMikuDance) とは、プリセットされたキャラクターの 3D モデルを操作しコンピュータアニメーション

*1 動画コンテンツのアバターはカクタス氏が作成したグラフィック風素体モデルを使用した(ダウンロード元: <https://bowlroll.net/file/102215>)

表 1 動画コンテンツ再生スクリプトの属性と詳細
 Table 1 Attributes and details of video content playback script

コマンド	属性	属性の詳細
heading	text	タイトルとなる文字列を指定する
	filename	タイトル表示時に描画する画像名を指定する
talk	text	講義資料の説明文の文字列を指定する
	filename	説明文を表示する時に描画する画像名を指定する
wait	wait	コマンド間で待機させる時間を指定する

ンを作成する 3DCG ソフトウェアである [8]. Three.js の THREE.MMDLoader ライブラリを用いることで MMD を使用することができる. THREE.MMDLoader では 3 種類の MMD のデータを扱える. 1 つ目はモデルデータの PMD と PMX である. PMX は PMD を拡張しており PMD より高機能である. 2 つ目はポーズデータの VPD である. これはモデルデータのポーズ情報を扱う. 3 つ目はモーションデータの VMD である. これはモデルや表情などのアニメーション情報を扱う.

動画コンテンツの描画は HTML で作成した Web ページ上で行われるため, OS やブラウザに依存することなく再生できる. これらの動画コンテンツ生成の機能はすべて JavaScript で実行される. 従来の動画コンテンツの提供では, サーバ上に容量の大きい動画ファイルをアップロードする必要があった. 本研究で開発するフレームワークでは, 動画ファイルに比べ容量の少ないファイルをアップロードするだけで済むため, 動画コンテンツ提供者にとって本フレームワークの導入が容易である.

図 5 に動画コンテンツの再生画面を示す. 学生が動画コンテンツの組み込まれた Web ページにアクセスすると図 5 の画面が表示される. 画像表示部には講義資料の画像を表示する. 説明文表示部では講義資料の説明文を表示する. アバター表示部では先生役のアバターを表示する. アバター表示部のアバターには動きを加えることが可能である. 現状では, アバターには左右に回転させるなどの簡単な動きをつけることができる. 将来的には MMD のポーズデータやモーションデータ等を活用して, 説明している箇所の指差しや音声と同期して口を動かす, 重要な説明箇所では閲覧している学生の注意を引くような動作を加えていく. また動画コンテンツは再生・一時停止を行うことができる. 図 6 に再生ボタンと一時停止ボタンの画面を示す. 再生・一時停止を行うには図 6 の左上にあるボタンをクリックする.

4.5 動画コンテンツ再生スクリプト

動画コンテンツを描画するために必要となるのが動画コンテンツ再生スクリプトである. 動画コンテンツ再生スクリプトは, 動画コンテンツの組み込まれた Web ページにアクセスした段階で動的に生成される. また動画コンテン

ツ再生スクリプトはコマンドとそれに対応した属性を持っている. スクリプトは以下のように記述する.

- コマンド名 (arg1 = data1, arg2 = data2 , arg3 = data3,)

属性はコマンドの引数に渡して使用する. また属性はコマンドによって種類や数が異なる.

次に動画コンテンツ再生スクリプトの一覧を以下に列挙する.

- heading コマンド
- talk コマンド
- wait コマンド

heading コマンドは説明する内容のタイトルを描画する. talk コマンドはタイトルに対応した説明内容を描画する. wait コマンドではコマンド間の待機時間を設定する.

表 1 に動画コンテンツ再生スクリプトの属性と説明を示す. heading コマンドの text 属性にタイトルとなる文字列を指定する. filename 属性にはタイトル表示時に描画する画像名を指定する. talk コマンドの text 属性には講義資料の説明文の文字列を指定する. filename 属性には説明文を表示する時に描画する画像名を指定する. wait コマンドの time 属性ではコマンド間で待機させる時間を秒単位で指定する.

5. 評価

本研究では動画コンテンツ提供のためのフレームワークを開発している. そのため実際に動画コンテンツの生成から提供までを行えるかを確認するための評価を行った. また動画コンテンツ提供の際には動画コンテンツ群を生成する必要がある. そこで動画コンテンツ群を生成するために必要な要素が利用できるかの確認をするために, プログラミング科目を例に初級, 中級, 上級のレベル別の動画コンテンツ群を作成した.

5.1 対象

評価にあたり, 使用する教材は東京電機大学システムデザイン工学部情報システム工学科で開講されているコンピュータプログラミングⅠ・Ⅱ・Ⅲの3科目を対象とした. コンピュータプログラミングⅠ・Ⅱは Java 言語を例に手続き型プログラミングを, コンピュータプログラミングⅢは

初級
<ul style="list-style-type: none">• 第1章 基本的な用語と操作• 第2章 オブジェクト図とクラス図• 第3章 DiceとArrayListクラス• 第4章 ArrayListとクラス• 第5章 DiceGameとインタフェース• 第6章 インタフェースとCoin• 第7章 シーケンス図• 第8章 ビンゴゲーム• 第9章 継承• 第10章 MobileTelephone• 第11章 ストリーム
第1章 基本的な用語
<ol style="list-style-type: none">1. 基本的な用語2. Jarファイルの解凍
第2章 オブジェクト図とクラス図
<ol style="list-style-type: none">1. 複数のオブジェクトの生成2. 演習2-1の解答3. クラスとクラス図4. Student表とオブジェクト5. オブジェクトの生成とオブジェクト図6. 演習2-2の解答と演習2-37. 演習2-28. ソースコードのクラス図からの導出9. 演習2-3の解答10. 未完成なクラス211. オブジェクト指向言語と手続き型言語12. newキーワードとオブジェクト13. 未完成なクラス114. 演習2-4の解答15. クラス図の変更と課題2-4

図 7 コンピュータプログラミング III の初級インデックスページの一例

Fig. 7 An example of an introductory index page of computer programming III

オブジェクト指向プログラミングを学ぶ科目である。コンピュータプログラミング I・II は 2 科目合わせて第 1 章から第 12 章までであり、コンテンツの総数は 121 である。コンピュータプログラミング III は第 1 章から第 11 章までであり、コンテンツの総数は 116 である。なおコンテンツの総数はまとめページを除外した数となっている。これはまとめページが講義資料内で使用された用語の補足事項になっているためである。

5.2 動画コンテンツ提供までの流れ

動画コンテンツを生成するのに必要となるのは講義資料とその資料で使われる画像である。コンピュータプログラミング I・II・III の科目では Web 上に講義資料が公開されている。

評価にあたり、実際に動画コンテンツを提供するまでの流れを説明する。はじめに講義資料の HTML ファイルと画像ファイルを用意する。講義資料は HTML 形式であるが、処理の際に XHTML に変換し使用する。もとの講義資

コンピュータプログラミングIII
予習・復習
コンテンツ
初級
中級
上級

図 8 コンピュータプログラミング III のレベル別インデックスページ
Fig. 8 Index page by level of computer programming III

料のままでは動画コンテンツ生成に関係のない情報も含まれているため、XML 形式の講義資料生成プログラムを用いて必要な情報に絞り、XML 形式の講義資料を生成する。しかし XML 形式の講義資料を生成した段階では動画コンテンツ生成に必要な情報のみが含まれているため、動画コンテンツ群の生成に必要な要素を設定する。今回の評価において、type, previous, next, keyword の要素は手動で設定した。type 要素は対象となるコンテンツが予習、復習、演習、演習解答のどの種類に分類されるかを人が判別し設定する。previous 要素は前のコンテンツ名を、next 要素は次のコンテンツ名を設定する。keyword 要素はそのコンテンツ内で重要なキーワードを手動で設定する。動画コンテンツの振り分け基準は初級はすべてのコンテンツを使い、中級では復習に該当するコンテンツは除外し、上級では中級の条件に加え演習解答を除外するとした。その後、この振り分け基準と XML 形式の講義資料を用いてコンテンツ振り分け XML を作成する。

インデックスページは初級、中級、上級の 3 つの HTML ファイルをあらかじめ作成する。HTML ファイルではコンテンツ振り分け XML の情報をもとにコンテンツを表示する JavaScript のプログラムを記述する。これによりコンテンツ振り分け XML の要素の値を変更するだけで容易にコンテンツ群を作成することができる。図 7 にコンピュータプログラミング III の初級インデックスページの一例を示す。このインデックスページはコンピュータプログラミング III の講義資料のインデックスページと共通の構成となっている。Web 上に公開されているコンピュータプログラミング III の講義ページと違う点は、コンテンツのリンクをクリックすると動画コンテンツが再生される点である。中級、上級も同じ構成になっているが、コンテンツの数はコンテンツ振り分け XML によって異なる。

図 8 にコンピュータプログラミング III のレベル別インデックスページを示す。初級、中級、上級のインデックスページは図 8 のようにさらにインデックスページにまとめ、このページからそれぞれのレベルのインデックスペー

表 2 コンピュータプログラミング I・II のレベル別コンテンツ数
Table 2 Number of contents by level of computer programming I・II

レベル	コンテンツ数	全コンテンツ数からの割合
初級	121	100%
中級	118	97.5%
上級	99	81.8%

表 3 コンピュータプログラミング III のレベル別コンテンツ数
Table 3 Number of contents by level of computer programming III

レベル	コンテンツ数	全コンテンツ数からの割合
初級	116	100%
中級	103	88.8%
上級	90	77.6%

にアクセスする。

5.3 要素を利用したコンテンツ群振り分け結果と考察

今回自ら振り分け基準を設定し、動画コンテンツ群生成に必要な要素を利用した。初級、中級、上級でのコンテンツ数に差が出ており、要素が活用できた。表 2 にコンピュータプログラミング I・II のレベル別コンテンツ数を示す。コンピュータプログラミング I・II の全コンテンツ数は 121 であり、そのうち中級は 118 個、上級では 99 個に減っていた。初級と中級のコンテンツ数にはあまり差がないが、上級では差が出ていた。表 3 にコンピュータプログラミング III のレベル別コンテンツ数を示す。コンピュータプログラミング III の全コンテンツ数は 116 であり、中級で 103 個、上級で 90 個であった。コンピュータプログラミング III では初級、中級、上級でそれぞれ約 10% コンテンツ数が減っていた。

次に要素を利用したコンテンツ群振り分け結果の考察を述べる。コンピュータプログラミング I・II でコンテンツ数に差があまりなかったのは新しく学ぶ項目が多く、予習に該当するコンテンツが多かったからではないかと考えられる。コンピュータプログラミング III では、コンピュータプログラミング I・II をベースに学んでいくため復習に該当するコンテンツが増えたからではないかと考えられる。

6. 動作実験

さまざまなコンピュータ環境で動画コンテンツが実際に描画され、閲覧できるか確認することを目的に動作実験を行った。2018 年 11 月 14 日に参加者 23 人を対象に動作実験を行った。動画コンテンツの題材として、東京電機大学システムデザイン工学部システムデザイン工学科で開講されているコンピュータプログラミング I・II を用いた。

動作実験にあたり、第 5 章の評価で作成した動画コンテンツのうちコンピュータプログラミング I・II の初級のイ

表 4 アンケート項目
Table 4 Questionnaire item

番号	質問内容
Q1	あなたのパソコンの OS を教えてください。
Q2	使用したブラウザを教えてください。
Q3	動画は再生できましたか？
Q4	音声は流れましたか？
Q5	画像は表示されましたか？
Q6	アバターは表示されましたか？
Q7	説明文は表示されましたか？
Q8	再生・一時停止機能は使用できましたか？
Q9	不具合があれば記述してください。

表 5 使用したブラウザの種類と人数
Table 5 Type and number of browsers used

ブラウザ名	人数
Chrome	18 人
Firefox	2 人
Opera	1 人
Safari	1 人
その他	1 人

ンデックスページを Web 上に公開した。実験参加者には、はじめはこちらで指示した動画コンテンツを閲覧してもらい、見終わった人には他の動画コンテンツを自由に閲覧してもらった。また使用ブラウザは任意としたが、Microsoft の Internet Explorer と Edge については動作確認が取れていないため除外した。

動画コンテンツ閲覧後に動画コンテンツに関するアンケートに回答してもらった。表 4 にアンケート項目を示す。Q1 と Q2 では参加者の実行環境を答えてもらい、それ以降の質問では動画コンテンツに関する質問に回答してもらった。

アンケートの回答結果として参加者のパソコンの OS は WindowsOS が 19 人、macOS が 4 人だった。表 5 に使用したブラウザの種類と人数を示す。使用ブラウザは Chrome を使用した参加者が 18 人と最も多く、Firefox が 2 人、Opera が 1 人、Safari が 1 人であった。その他は 1 人で、使用したブラウザは vivaldi であった。動画コンテンツに関する質問の回答では、動画コンテンツを再生できない、音声が流れない、画像が表示されない、アバターが表示されない人はいなかった。また再生・一時停止機能を使用できない人もいなかった。不具合の自由記述においては、「動画コンテンツの再生途中でブラウザの処理が停止し動画コンテンツが止まってしまう」、「ブラウザの画面が途中で黒くなり、音声のみが流れ続ける」などの記述があった。また「ソースコード」と読むべきところが「ソスコド」と読み上げられ、文章の読み上げがおかしな部分があった。

6.1 考察

動画コンテンツの描画処理はすべてクライアント側で行われる。動画コンテンツは Web ブラウザ上に 3D モデルのアバターを描画しており、パソコンの処理性能が要求されるため今回の動作実験において途中で処理が停止してしまう不具合が発生したと考えられる。また Three.js は WebGL を容易に扱うことができるライブラリであるが、この Three.js の処理そのものがパソコンに負荷をかけていることも考えられる。今回の動作実験で生じた不具合が起きないように、さまざまなコンピュータ環境においても安定して動画コンテンツを閲覧できるように修正していく必要がある。

文章の読み上げがおかしくなっていた部分に関しては、今回の動作実験でプログラミング科目を教材として使用しており、この教材に出てくる単語が一般的な単語ではなかったため読み上げがおかしくなっていたのではないかと考えられる。教材に出てくる単語の読み上げを正しく行わせるには単語の辞書等を作成し、適用していく必要があると考える。

7. まとめ

本研究では学生に合った予習と復習用の動画コンテンツを提供するためのフレームワークを開発した。このフレームワークを用いることで動画コンテンツを容易に作成し、学生に合った動画コンテンツの提供を行える。学生に合った動画コンテンツを提供するために必要な要素を電子化した教材に含ませたことで、動画コンテンツ群を授業や課題ごとにコンテンツの振り分け基準を変更して提供することができる。

開発したフレームワークが提供する機能が利用できるかを確認するための評価を行い、動画コンテンツの生成と動画コンテンツ群生成に必要な要素が利用できることを確認した。動画コンテンツ群生成に必要な要素を用いて初級、中級、上級に分類した際に、コンピュータプログラミング III では難易度が上がるにつれてコンテンツ数が約 10% ずつ減少していた。

動作実験として 23 名の学生に動画コンテンツを閲覧してもらい、再生できることを確認した。また動作実験の中で、動画コンテンツの再生途中でブラウザの処理が停止してしまうなどの不具合を確認した。

8. 今後の課題

学生に合った動画コンテンツ群の生成に必要な要素について引き続き検討する。現状ではアバターの動きがあまりないため、読み上げている箇所の指差しをするなどのさまざまな動きにも対応できるようにする。今回の評価においてはプログラミング科目を対象にフレームワークが利用できるか確認したが、今後は別の科目においても適用で

きるか確認していく。

動作実験で生じた不具合が起きないように、さまざまなコンピュータ環境においても安定して動画コンテンツを閲覧できるように修正する。また動画コンテンツを閲覧する際に Microsoft の Edge 等のブラウザでは動作確認が取れなかったため、動作確認の取れなかったブラウザでも閲覧できるようにする。

参考文献

- [1] e ラーニング戦略研究所：ビデオ教材（映像コンテンツ）の教育利用に関する意識調査報告書，デジタル・ナレッジ（2015）.
<https://www.digital-knowledge.co.jp/archives/1623/>, (参照 2018-10-16)
- [2] 中央教育審議会：大学教育部会の審議のまとめについて（素案），文部科学省（2013）.
http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo4/015/gijiroku/1318230.htm, (参照 2018-10-16)
- [3] 濱中 義隆：平成 28 年度 大学生等の学習状況に関する調査研究，国立教育政策研究所（2014）.
http://www.nier.go.jp/05_kenkyu_seika/pdf_digest_h29/gaiyou.pdf, (参照 2018-10-16)
- [4] 大学 ICT 推進協議会（AXIES）ICT 利活用調査部会：高等教育機関における ICT の利活用に関する調査研究 結果報告書（第 3 版），大学 ICT 推進協議会（2015）.
<https://axies.jp/ja/ict/2015report.pdf>, (参照 2018-10-16)
- [5] 林 正樹，折原 豊，下田 茂，上田 博唯，横山 敏明，八重樫 一仁，栗原 恒弥，安村 通晃：テレビ番組記述言語 TVML の言語仕様と CG 記述方法，第 3 回知能情報メディアシンポジウム，pp.75-80（1997）.
- [6] 市村 哲，福井 登志也，井上 亮文，松下 温：Web 学習用講義コンテンツを自動作成する板書講義収録システム，情報処理学会論文誌，Vol.47, No.10, pp.2938-2946（2006）.
- [7] 八重樫 理人，谷川 晃，守屋 英樹，玉田 裕司，神澤 雄智，三好 匠，相場 亮：講義コンテンツ自動生成システムの開発，電子情報通信学会論文誌，Vol. J91-D, No.12, pp.2819-2832（2008）.
- [8] Jos Dirksen（あんどうやすし訳）：初めての Three.js 第 2 版，p.416, O'Reilly Japan, 東京（2016）.