

This article is an accepted extended-abstract of Digital Humanities 2020 (DH2020: <https://dh2020.adho.org/>). Due to the pandemic of Covid-19 or Chinese Virus called by some people, the in-person conference was cancelled. This paper handles an online course simple and easy to make. Thus, on my own decision, I publish this as a pre-print. The whole source code can be downloaded from the following URL.

- <https://docsci.infon.org/stack/LfR/>
- <https://docsci.infon.org/stack/LfR.zip>

An online course system easy to make, preserve, and promote critical thinking

Kazushi Ohya
Tsurumi University

In this poster presentation we will show a new online course consisting of talks and chalks that is easy to make, edit, and preserve, and is a traditional and old-fashioned lecture style, that substantially helps students learn the content spontaneously.

Delivering, sharing, or learning new knowledge through the internet is the new common style for every generation. We scholars also are expected to expose our knowledge to learners as fast as we reasonably can. The present online courses like Moocs are made with very stimulative and attractive media or content with the results of CAI or edutainment research. However, this kind of high-stimulative content may deprive students of time to actively consider the topics. Too attractive content causes students to immerse themselves in the lectures and consequently could make them passive learners. And, this type of content requires a massive amount of time to make it, which means the cost of making the content is expensive and it is not easy for educators to find the time and chance to make it, thus it is difficult to be issued timely to students, therefore the number of the courses can not be expected. Worse still, the present contents of online courses are hard to be preserved on a common platform such as archives because of heavy-dependence on specific software or systems.

To resolve the above drawbacks, we designed online-courses with a simple style that mimics real class teaching such as a radio lecture with slides. The course materials we need to make are only scripts for talks and XML data for slides, and the process we need to undertake separately is just recording the talk. The text materials are typed in XML data(Figure 1), and talks are recorded in a common sound file such as an mp3 that can be realized on a web browser with the function audio in JavaScript(Figure 2). The mechanisms to realize a slide show are backed with HTML5 and simple codes in JavaScript(Figure 3 and 6). The XML and sound data are system-independent data formats, or robust digital objects for preservation. The moderate attractiveness this lecture style provides could be satisfying for active learners. Using this system we have opened a course of a data model for DH students and linguists engaged in language documentation[1][2].

The technique adopted in this system comes from the specification of HTML Imports[3] in Web Components[4]. Unfortunately the HTML Imports would be obsolete in the near future[5][6]. However, fortunately, Google has continued to endeavor to complete the specification[3] and provide JavaScript libraries webcomponents.js for using the functions of HTML Imports for the time being[7]. This lecture system is important not only for an implementation to realize a new teaching/learning channel conveying the knowledge instantly to learners but also to show the evidence for usefulness of HTML Imports as a sign of the existence of users' requirements.

```

<?xml version="1.0"?>
<slides>
  <div id="ja">
    <div class="slide">
      <h1>第 5 回 : Data Format</h1>
      <ul>
        <li>Data Format; データ形式</li>
        <li>情報を計算機上のデータとして記録するためのルール</li>
        <li>ワードデータの doc/xdoc , 写真の Jpeg, 音声の Wave/Wav や mp3 など . </li>
        <li>普段の生活では , ファイルの拡張子にデータ形式がみえる . </li>
      </ul>
    </div>
    <div class="slide">...

```

Figure 1: slides in XML Data

```

function playSound(lang){
  if(lang == 1){
    audio.src = audioJ;
  }...
  audio.currentTime = 0;
  audio.load();
  audio.play(); ...

```

Figure 2: playing sound data in JavaScript

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Lectures for Radio on RDS</title>
    <link rel="import" href="./slides.xml" type="text/xml"></link>
    ...

```

Figure 3: an element link in HTML5 head

```

function readXML(i){
  let doc = document.querySelector('link[rel="import"]');
  if(i == 1){
    contAll = doc.import;
    cont = contAll.querySelector('#ja');
  }...
  let slides = cont.querySelectorAll('div');
  loop(slides,i); ...

```

Figure 4: a function import of HTML Import

Lectures for Radio

consisting of talks and slides for the coming Radio Data System

A list of lectures

Data Model

1. Introduction
2. System Engineering
3. Function First vs Data First
4. Data, Function, and View
5. A bundle of relations
6. Primitive Relation
7. Record
8. Table(regular)
9. Table(interleaf)
10. Tree
11. Graph
12. Your own data

A Bundle of Relations

1 / 5

再生 停止 play stop

第5回 : Data Format

- Data Format; データ形式
- 情報を計算機上のデータとして記録するためのルール
- ワードデータのdoc/xdoc, 写真のJpeg, 音声のWave/Wavやmp3など.
- 普段の生活では, ファイルの拡張子にデータ形式がみえる.

Figure 5: a web page made of the XML data in Figure 1

Lectures for Radio

consisting of talks and slides for the coming Radio Data System

A list of lectures

Data Model

1. Introduction
2. System Engineering
3. Function First vs Data First
4. Data, Function, and View
5. A bundle of relations
6. Primitive Relation
7. Record
8. Table(regular)
9. Table(interleaf)
10. Tree
11. Graph
12. Your own data

Interleaf Table

8 / 10

再生 停止 play stop

ToolBoxによるデータの例3

..

¥u We accepted their terms.

¥ps pron. verb pron. noun

¥cs sub. pred. poss. acc.

¥mr pron. verb+past pron. noun+pl

¥mn わたしたち 受け入れる+過去 それらの条件+複数

¥jp それらの条件を受け入れた.

¥u I lunged forward to try to hit him

¥ps pron. prov. prop. pron. verb pron.verb pron.

¥cs sub. verb adv. prop. verb prop.verb pron.

¥mr pron. pred.+past fore objective pred. prop.pred. pron.

¥mn わたし 出る+過去 前に ために 試みる null 当てる 彼を

¥jp 彼につきを入れるために前に出た.

¥uが出現する毎に, データの単位が変化してゆく(異なっている).

レコードは, ¥uを中心に, まとまりを作っている.

この情報は何処にも定義されていない.

→定義されていないものはデータ処理ができない.

もしくは, どこかにこれらの定義を書いて置かなくてはいけない(どこ?).

↓

Interleaf Tableは, 言語資料のデータ構造としては都合が悪い.

Figure 6: a more complex slide

References

- [1] Ohya, K. (2017) “Two Main Contents in a Syllabus for Language Documentation: the Learning Data Models and an Assignment of Data Conversion”, The 5th International Conference of Language Documentation and Conservation (ICLDC5), <https://scholarspace.manoa.hawaii.edu/handle/10125/42045>, University of Hawaii
- [2] Online Course on Data Model for Language Documentation, https://infon.org/lr/lddm/lddm_top.html
- [3] HTML Imports W3C Editor’s Draft , <http://w3c.github.io/webcomponents/spec/imports/>
- [4] Introduction to Web Components <https://www.w3.org/TR/components-intro/>
- [5] Can I use HTML Imports? <https://caniuse.com/imports>
- [6] HTML Imports https://developer.mozilla.org/en-US/docs/Web/Web_Components/HTML_Imports
- [7] webcomponents.js <https://github.com/webcomponents/webcomponentsjs/tree/v1>