

Modelling an interactive Text editor.

1363190

Abstract

This paper Evaluates the text editor model, By making use of audio and video player, Line calculation and cache storing and definition allocation . The model shows the possibilities of Empirical modelling In jseden by use of dependencies, observables, and other Definitions, This paper reviews the use of Empirical modelling in the Visual and informative methods based on understandable and clear design. The process to gain the results, the limitations and the benefits of Empirical modelling. Further possibilities that can grow from EM and a personal conclusion to if Empirical modelling is a tool that accomplishes what is was designed to do.

1 Introduction

To Begin, ill give a more basic explanation of what was to be the achievement and shall give that conclusion at a latter stage. The aim was to create a tool that could be used in a education related environment such as in a conference. The tool works by links. If time allowed would have been operated primary by system inputs such as microphones on laptops and cameras for recording such as with a rare facing camera. By allowing you to take notes simultaneously recording the audio to then re-view afterwards. This was the original idea but the idea was altered as it commenced.

To gain the designed tool the Model needed the following

- Video Input
- Audio Input
- Text Input which operated the Audio/Video time line
- Called functions used to store bits of data.

The Main purpose of this was to create a educational fronted tool that made use of graphics and relied on inputs rather than a constructed simulation. The second purpose was to show the ease of written language to allow the use of

other components such as HTML elements and mp3 players.

1.1 The models interface.

This being the front end of the model, needed to be easy to use and clean(by which I mean simple. By making use of html and DIV functions the layout remains clear and those replies on calls by the user. Such as shown in Figure 1. By having a html function wrapped around a DIV it allows for links to audio to be attached through a Button names "P1, P2,P3"these remain in their locations until replaced by a new link is, designated to the buttons. Although this wasn't the ideal design compared to the proposed input recording it seemed more logical as recording can be gathered from sources like drop-down list, (image1) and other servers [Dropbox]

Bit of audio became useful as its buffering allowed for storage of multiple links. The EM design behind this allows for simple reading by most as its clear as what funcntion does whatt.

For Image 1 This was its code which created a dependence between the Input-box, Buttons and Player.

```
HereisAudio is Div("mysound",left,410,0,0,"<audio id='my-actualaudio' controls><source src=' ' // AudioLink // '" type='audio/mp3'/></audio>");
inputsource is Inputbox ("linkhere",left, 210, true);
Assbut1 is Button("AssBut1","P1",left+10 , 240, true);
proc Getlink1 : AssBut1_clicked {
    Get1=linkhere_value;}
Play1 is Get1;
AudioLink is Get1;
```

This method was used several times for adding values to a dropdown list which meant that the list was ever changing, if needed. Being able to constantly store values on "Current values or against definitions means that users can experiment more with possibles. This same principle can be used all through out this model which allows more interaction, As the model can better be understood. [BretV]

1.1.1 The use of players

The use of Video and Audio player has some keys importance as its intended tool. Being able to take notes, listen to what's happening, record the visual and audio aspect and have it all synchronized together is fundamental to gaining a better and more useful tool. The aim was to have a constantly recording tool which rules through the buffer memory[w3school] until the record button is pressed. Being able to allocate the amount of pre-recorded to then add to the file as well as a tic tool which runs the same method as citing a paper but instead the user is citing the locations of the recording. Calling the .currentTime function, the user is able to use the tool how they feel. Example for a simple marker placing and calling.

```
currentTimeNow is myactualaudiojse.currentTime;
EntireDuration is myactualaudiojse.duration;
Citehere is Button("cite", "Mark", 230,170, true);
proc Citing: cite_clicked{
```

```
Cite1=currentTimeNow;
Citehere2 is Button("cite2", "Mark", Citehere.x1,Citehere.y1, true);}
Go002 is Button("Go00021", "Cite1", 300,100, true);
proc movetime : Go00021_clicked {
    myactualaudiojse.currentTime=Cite1;}
```

1.1.2 Areas of interest...

implementation of a clock which allows for recording of the marker button [EM]s it relies on the tic+ function this was collected from a JS called function. With that said this model drags on the interaction between the user and the observables.

As an educational Tool, Empirical modelling is very useful it can be argued although this is down the personal opinion. The use of dependencies and other controls enhance the learning with the abilities to create and understand, by the use of tools such as the Dependence modelling. Figure 3 shows the links created in the text editor model. As mentioned by [Russ, 1997] most current tools are rigid less forgiven on system errors this is not to say that jseden is easier than others, but as a tool for learning and creating possibility is very useful to this day and age.

A part Figure of Dependency tree:

nt of EM is the dependence. The Text editor model make use of multiple dependence such as in figure 2, Being able to modify exact areas of the models without large list of further adjustment is not only beneficial but also productive and encourages more use and experiments. A benefit of multiple dependences is being able to link values and have them complete different task without invading other values as such with html.

```
audioLine1 is Line (left+10, 350,right-10,350,"blue");
audioLine2 is Line (left+10, 355,right-10,355, "yellow");
```

Figure.3

the area of the box the benefit of this, is that it allows for instant adjustment by the function if the box weight is controlled by the canvas, having the canvas recorded its possible to then assigning a value to the canvas x and y values in this case r was for the Wight of the box. Which then placed with added value to create the ever correct line. where the line is changes. Chains of Dependences means that the controls adjustment can be made.

2.1.2JSEden, tkEden and Others

A limitation with jseden compared to Eden and other platforms is grouping at current jseden isn't able to group lines or values together, together so for example to draw a symbol such as the plus sign normally 3 lines would be used, but jseden does not operate a grouping function is can be argued that it can be created by individual lines but this values cant become fixed instead are only allowed to their pixel point not their subgroups but can be located to certain points according to the canvas side. Which in this case I labelled "right"

Other areas of downfall is the lack of an internal reset being able to clear a workspace and or symbol list with the site would be very useful, at current resetting or creating a new site is only

possibly refreshing or resetting the page, it seems a caching problem is also apparent in jseden unlike the Eden which has the ability to clear its cache, jseden is linked to the browsers bin and can sometime carry an error. As shown in figure 4.



Figure 4

It can be argued that the jseden has its good points which I would also argue on being able to write definitions in a single file . An area that Jseden worked effectively is in being able to call videos from their html locations.

Figure 5 shows the console locations and the embedded player, this in most cases are located under the java script shown in figure 5.

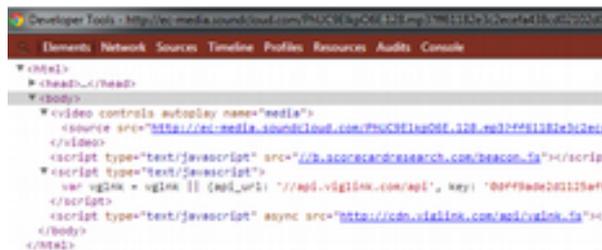


figure 5

A strong benefit to Jseden is being able to call back previous functions and alter them in a new input. Although the layout is weak the

writeln()

function allows for quick preview of values. Being able see if changes have been made by its use. This writeln tool is also implemented in tkeden although tkeden relies on scout, Donald and Eden separated grouping what working on project you are able to see if changes have been made .

3. Player and Link

3.1 The uses

The use of video and audio players follows the design used by [E Hudnott] having dependencies which rely on watched observables to call html such such as :

```
$$  
  
declare_jse("myactualaudiojse");  
myactualaudiojse = myactualaudio;  
  
}}$; [E Hudnott]
```

This script places a html tags such as :

```
myactualaudiojse.currentTime=X  
X= myactualaudiojse.duration  
myactualaudiojse.goplay=X
```

X being the value and .html element being the html tag being gathered a list of possible html elements can be sourced from the internet [w3school]. Creating a player within a DIV and a link through a value creates dependences between which can then be added to create or trigger other functions. The use in my model was to assign values to symbols. Making use of symbols these allows for dependence to remain the same but the values link to the dependences can constantly change such as the Audio link and Video value. Shown in below from the text editor model:

```
mySuperVideo is  
Div("myvideo",right-270,20 ,10,right-  
50,"<video id='myvideo'  
controls><source src="" // VideoIn-  
sert //  
"type='video/mp4'/></video>");  
WriteInsert is Inputbox  
("videoln",right-270, 210, true);  
VideoInsert is videoln_value;
```

The use of two different functions linked together by the use of *VideoInsert is videoln_value*;

For each player its important to make use of the link type.

```
type='video/mp4'
```

specifying what type of source is being used allows for the correct internal functions to be use, The Text editor models makes use of mp3 and mp4, which can be alternated without affection other components of the programme, This is a good explanation of EM, Having a strong understandable of the observables instead of having it mathematically describe.

3.2 Empirically modelling.

Empirical modelling is an “ interaction with programs”[EM] and this models is able to demonstrate areas of what Empirical modelling sets out to accomplish, as it brings the front facing interaction of computing with everyday computing such as being able to place links and text tools used in everyday situation into a more productive tool .

An inbuilt audio player although correctly a front end to a html player shows what can be done and if time allowed could grow into a more power and easier tool for users to use. Its easy learning curve means that Java script uses and other language uses can read hows files and observables and labels and build an understanding of what the programme does and what it can do.

4.Evaluating Empirical Modelling

4.1

Working by EM allows for better understanding of a model. It gives large areas for development, adjustment of observables and dependences

Subheading 4.2 gives an evaluating of the tool in creating a prototype for an interactive text editor.

4.2Evaluating the Model

The model which first set out to archive a test were for writing and linking audio to text accomplished and then improved by having a

video player and buttons to call cite location, time control, pause and play functions as a front-end to HTML audio and video elements.

As The models stands it is only a prototype and shows a strong understanding of the dependences and observables as shown the DM image. Being able to read through work and have the audio synchronized to it simultaneously as being able to call by exact location marked in the text is an idea that can be improved on. But with other time constraints couldn't improve on this idea in time. Future work will be mentioned in the subheading- 4.3.

The models design was lacking in quality as a prototype it was to show how and what can be done. Although not perfectly synchronized as it runs through value calculation and with line value it , it has room for improvement and better gaining of knowledge.

Another slight problem with it seems the input ran or updated it self in different ways earth time, so have a constantly problem with ways or writing symbols such as < and > .



Overall this model made use of new functions not done in Eden or Js eden before this was a new approach which I set to be different from the others. This use is as better and now can allow for new changes and alternations with less limits in what the users. Empirical modelling as a whole is also a much better teaching approach than conventional mathematical computations. I think this is better and although the model is not the most visual pleasing tool with its lack of labels, it does what is was set out and made use of new functions such as the DIV and and subbing links.

The use of Time fuction although runs on its own, has a refresh error when added with other dependences, this problem meant it couldnt be added to the marker and was commented out of the over project. Which meant that the marker didn't give a cite with time attached to it.

Future works

As the model stand it has several areas for improvement from design and layout to function . The original aim i believe has been accomplished, The implementing of line to sync between the text and the audio.

Areas such as the Video player and audio area the model has shown can be implemented in jseden and this process can also be defined in tkEden. The largest area of improvement can be the use of better interaction and layout an attempted use for improvement of button was having a mouse pressed function which would call on exact location to then give a result. Figure 7 shows an attempted code for it. Although it wasn't completed it gives what can be done in Jseden and doesn't reply on other factor other than the pixels locations.

Having a button for cite than can print out new values when processing an area of improvement but a downside to this would be the amount of control and values it results it. This model can be continued by most as the language design is clear to understand improvement on this model is possible and its a good example of the dependencies that extend beyond the perimeters of jseden.

Conclusion

Empirical modelling although new to it, is very effect and useful with many situation being able to understand at similar level to English language means that most can understand it, The use language can be applied to all or most language. coding and concept learning will benefit the users who aim to improve on their way of computing.

I feel that my model gave a strong explanation on dependencies and there abilities although as mentioned it being a prototype the ideas have been created and now needs to be modified to clean and work

properly as a prototype it archive its desired results and gained new possibles in the video field. This process as mentioned in the sub heading future works it to be a more interactive and more productivity curve to users who learn empirical modelling at it not gives results but also shows possibilities The model set out to show the dependences between text and audio, as this was archived the further work with

video, droplist and buttons shows that Empirical modelling and Jsedn not only gives shows new experimental methods but helps in informative ways.

Acknowledgements

I would like to thank Dr. Meurig Beynon for his informative lectures, phd students who helped in given explanations in labs and Fellow students for opening questions and new logical reasoning

REFERENCE

BretV: Bret Victor, Transcript of “Inventing on Principle”, 2012

Russ, 1997: S Russ, Empirical modelling: the computer as a modelling medium. , 1997

E Hudnott: Elizabeth Hudnott, audioHTML5test.js-e, 2013

w3school: http://www.w3schools.com/tags/tag_audio.asp

EM:] M. Beynon, (2013) Empirical Modelling for Educational Technology. University of Warwick, Coventry.