This presentation will draw your attention to an online web application for viewing worked examples. It is aimed to be used in schools across the UK.

**What are worked examples?**

Let me first introduce you to the concept of worked examples.

A worked example is an example that is broken into different pieces. These pieces are gradually revealed to the reader as a step-by-step process. Each step has explanation of the logic involved in it.

**Why are worked examples beneficial for learning Computing Science?**

Judgement and decision-making play a huge role in solving Computing Science problems. However, novices often have a very restricted knowledge on the domain involved in a particular problem due to lack of experience. Worked examples can help them build up the necessary expertise required to solve a particular type of problem effectively.

**Problems of the existing forms of worked examples**

Books and lectures often provide worked examples. So why bother creating a tool for worked examples? Well, there are a few problems with the worked examples provided in books and lectures.

Firstly, the worked examples in books and lectures are not interactive enough. With these examples often the process of thinking why a particular action is undertaken or is a better option for reaching a solution remains unexplained.

Secondly, the worked examples present may not fit well enough to the teaching needs. Books aim to target a large portion of potential readers so they need to be general enough to fit every reader’s needs. However, this means that one particular reader may need to adjust their studying or teaching around this general example. What would be more beneficial is to adjust the worked examples depending on the teaching or learning needed.

Finally, worked examples in books provide little or no feedback on how they were used. Such feedback can help authors of examples create better ones as well as teachers using these examples to identify any difficulties their students may have.

**The solution to these problems**

As part of his research project, the former Glasgow University PhD student Dr. Yulun Song developed a Java standalone application to facilitate the creation and viewing of worked examples. This tool gathers usage data for the worked examples to be used by authors to improve these examples. This tool is called IWE standing for Interactive worked examples.

**The context of worked examples in this Level 4 project**

The outcome of this Level 4 project is intended to be a tool for worked examples to be used in schools across the UK. This tool is based on the ideas around the IWE tool and would be an attempt to improve the techniques used to teach Computing Science in the classroom.

**Why not use IWE in schools?**

Dr. Song’s prototype aims to prove that worked examples are beneficial as a technique for learning Computing Science. This prototype, however, was not aiming at deployment but it was rather an experimental tool. The fact that the prototype is a Java application brings in some complications for deploying it in schools. The reason for these complications comes from the fact that in schools across the UK there tends to be a blanket policy about the provision of software for any subject. This means that installing IWE in schools across the UK will most certainly turn into a time and cost-consuming task. This would be enough to prevent most teachers from considering adoption of the software.

**An alternative**

The issue of software provisioning in schools gives the major motivation for this project to recreate IWE as a web-based application in order to start effectively presenting worked examples in a larger context. This will avoid the complicated and time-consuming process of installing IWE in schools. The web-based version is called *Worked Examples Viewer*

**Other benefits**

Another benefit of the tool being web based is that schools will be able to receive the latest updates of the application and its worked examples with no effort. Furthermore, a web-based system could share worked examples developed nationally and even internationally, not only the examples created by one teacher.

**Taking a step further…**

In addition to being a more easily deployable version of IWE, WEAVE takes a step further to move from author-student to author-student-teacher target user groups. This brings in interesting new aspects. Teachers will be able to see personalised information about how their pupils interact with the examples. Authors, on the other hand, will receive information about the general usage of these examples, rather than personalised one.

**A necessary condition**

It has been decided that the main purpose of this project is only the provision of a student interface and a teacher interface for the new system. This essentially means that the system will need to use the worked examples created using the author interface of the old system. It is essential for the new system to be able to upload examples created by the IWE tool.

**Goals**

This project aims to achieve four goals:

- build a web-based viewing system that is interoperable with the author interface of IWE, i.e. ensure that worked examples created using the old system can be viewed in the new system.

- provide an interface for teachers that will help them gain more information on how the worked examples are used by their own pupils.

- replicate as closely as possible the student interface of IWE.

- ensure that worked example authors can view usage data in an anonymous manner, such that individual pupils, classes or schools are not identifiable.

**Architecture**

WEAVE is realised as a Django web application. Its architecture consists of three distinct tiers.

* Presentation tier. This is the top level of the overall architecture also known as the client side web interface. It defines the appearance of the website by rendering HTML and CSS and provides means for users to interact with the application. The clients are in the form of web browsers. On every interaction, they send requests to the server in the form of Ajax GET or POST requests to ensure asynchronous communication between the client and the server.
* Django Middleware. This tier consists of two distinct components.
  + The first component of this tier serves as a communication point between the client and the database where the examples and their usage data are stored. In this tier the requests from the client are parsed and translated into ORM requests- a language understandable by the database. These requests are passed forward to get or store the information in the request from/in the database. After the backend generates the response, the middleware is responsible for passing it back to the presentation tier in the form of Http response.
  + The second component is the connection point between IWE and WEAVE. This is where the translation of the XML elements from the old tool to ORM objects used for the new tool is taking place. This is done via an XML tree parser.
* Data layer. This tier represents the database in which all the information used or generated by WEAVE is stored in the form of objects. On GET requests the backend tier responds with an object meeting the criteria specified in the ORM request. On POST requests, the database creates a new object with the features specified in the request and stores it in the database.

**Teacher Interface**

**Main page**

The purpose of the teacher interface is to enable teachers to see usage data for their classes as well as individuals in these classes. I will show you how this is achieved using a screenshot of the main page. Teachers will register each of their classes as a group. They will then specify the number of students in the class they are creating. They can add more students to this group using the update option, as well as delete a group. Once the group is registered, the teacher can view the pupil ids which were automatically generated by WEAVE. Such automatic random generation of pupil ids is to ensure that the id does not reveal the pupil identity to avoid privacy issues.

**View Data**

In order to view data for their classes, the teacher needs to click on the view statistics option. Then they need to enter the details of the group they want to view data for, as well as the type of data they are interested in. The type of data usually is represented in the form of graphs and includes the average time spent by all students or the total time spent by a particular student at different steps of an example, as well as any answers to any questions.

**Student Interface**

One of the goals of this project is to translate the student interface of IWE into an online version. The research of Dr. Song clearly indicates that the user interface he uses for his system is effective at communicating the worked examples. I will illustrate how WEAVE’s interface incorporates the main features of the interface of IWE using screenshots of both systems. On the right hand side you can see the old system’s interface. The new system’s interface is on the left part of the screen.

Area one shows the worked examples installed on the system. The new system’s interface will show a dropdown list of examples when the user clicks on it- this saves space.

Area two has panels showing different documents for a particular worked example.

Area three shows the controls for the transitions between steps. Again, to save space, the bar showing the progress is removed. The same information in a neater form is shown as part of the explanation. As the student uses these controls to move through the worked example, the contents of the documents panels and the explanation area change to reveal the developing solution and the thinking process behind it.

Area four is the explanation area where the expert’s process of thinking involved on the current step is shown.

Highlighting of the newly appeared text at a particular step for drawing the user’s attention to the new content for the current step is shown in area five.

Highlighting of fragments of interest for a particular step is shown in area six.

Other features of the student interface, which are not shown on the screenshot, are the ability of the tool to ask the user questions and to record data such as time spent at each step and answers to questions.

**Evaluation**

There are four stages of evaluation taking place at the moment.

* Usability Evaluation conducted by participants who are not teachers or students in Computing Science. The reason for this choice of participants is so that the user interface can be judged with no bias. In other words, this evaluation aims to examine how easy to use and how self-explanatory the interface is.
* Heuristics Evaluation conducted by two experts in Django web application development- Dr Leif Azzopardi and Mr David Maxwell- authors of the award nominated book How to Tango with Django, which I followed in the stages of development of the system.
* Usability Evaluation with teachers in schools. They will be able to provide some feedback on how good the application is for the intended context as well as whether it has been achieved its goal to be easily deployable in schools
* Practical Evaluation by the secondary school teacher Mr. Peter Donaldson who will use it as a teaching technique with his students to evaluate WEAVE more thoroughly in achieving its goals, in other words- is it easy to use WEAVE in schools, is it well-accepted by students and is it helpful for him as a teacher to better understand the problems of his students.

**Future work**

After teachers have had the opportunity to work with this tool, more and more demands for different features of the application are expected.

In addition, there is a group of users whose needs are not addressed by the current tool- the authors. It is very difficult for authors of worked examples to create such examples via the author interface of IWE. Hence, the next stage of this project is to provide an easier web-based interface for this group of users too.

If successful, WEAVE is just the beginning of a tool which may be revolutionary in the area of teaching Computing Science in schools.