University of Glasgow | School of Computing Science

Level 4 Project Report

Emilia Vulpe

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March 31 2015

Abstract

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

Chapter 1

Introduction

Teaching programming is inherently difficult. The way it is taught in schools in the UK is by explaining the basic concepts related to a specific topic, presenting simple examples to illustrate how these concepts can be applied and posing a more complex problem for pupils to solve. However, in the initial stages of becoming programmers, beginners often lack a good enough understanding of the domain from just the simple examples to be able to solve the problem [6, 11]. This can lead to pupils struggling to find a solution, rather than gaining a better understanding of the problem-solving process.

A good way of teaching somebody an intellectual activity is by showing them the process of thinking involved in carrying it out. This is a form of apprenticeship known in the literature as "cognitive apprenticeship" [5]. In any apprenticeship model, the learner needs to see many examples of the activity to be learned in order to develop the experience necessary to attempt a new, related, activity. Unfortunately, due to limited number of hours dedicated to each individual subject in schools, teachers are somewhat restricted to using only the traditional methods of teaching. Time simply would not allow them personally to show their pupils many examples of what cognitive steps they should undertake in order to solve a problem.

Step-by-step guidance of the process of solving particular problems can help beginners gain a better understanding of the problem-solving process generally. Books provide such guidance in the form of worked examples. These examples have proven to be effective [15, 17, 20, 21]. It has also been found that they reduce the cognitive load when acquiring a skill [16]. However, such books may not necessarily accommodate the needs of a particular teacher. Furthermore, finding a close enough example for a particular topic may become a time-consuming and discouraging activity for a teacher.

Having this in mind, a Glasgow University PhD student, Yulun Song, has developed a Java standalone application called *Interactive Worked Examples (IWE)* [14]. It aims to address the issues mentioned above as well as to evaluate to what extent such an application will prove effective in lowering the learning curve for students in Computing Science. It consists of two interfaces: one for students and one for creators of worked examples (who are typically teachers as well). The author interface enables the creation of examples to accommodate a teachers specific needs. The student interface provides users with a selection of examples to work on.

The application has proven to be effective at enhancing the teaching of Computing Science in university [14]. Since the research questions around IWE were to explore the extent to which it can fit in the teaching process in a university context and whether it would be a potentially successful learning technique, the prototype does not aim at large scale deployment. A sensible next step is to put the system into use in schools, where support for computing education is urgently needed [13]. However, many issues in deploying IWE arise because of it being in the form of a Java standalone application. In schools in the UK there tends to be a blanket policy about the systems provision on any subject. In order to install a program on a school machine, a request to the service provider responsible for the particular school needs to be made. The service provider will then need to

analyse the risk that installing a new program will pose to the whole system and submit a further request to a local authority responsible for the particular school. This overhead would be enough to prevent most teachers from considering adoption of the software, both from a time and cost standpoint.

This 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. Furthermore, schools will be able to receive the latest updates of the application and its worked examples with no effort. Ultimately, a web-based system could share worked examples developed nationally and even internationally. The web-based version of IWE is called *Worked Examples Viewer (WEAVE)*.

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.

Another benefit of WEAVE being web-based is that the worked examples in the system will not be limited to the ones created by one teacher or a group of teachers only. Instead, examples created by any teacher will immediately be available to everyone. This would contribute to a collaborative way of developing such examples and would give the chance for pupils to undertake further learning if they desired so. Furthermore, teachers would be able to benefit from their colleagues expertise as well as get ideas and adjust them to their specific needs with less effort than creating new examples from scratch. Ideally, such a system can be revolutionary in improving the teaching practices in schools, help teachers understand the difficulties of their pupils and enable them to help each other to become better in teaching Computing Science.

The rest of this dissertation describes more background for the context of the project, the requirements for, as well as the design and the implementation of WEAVE together with the testing methods that were used to ensure that the application works as intended. An evaluation chapter follows making conclusions about how easy to use and to integrate successfully in everyday teaching practice WEAVE is. The final chapter is dedicated to the conclusion and the future developments for the system which will be addressed shortly.

Chapter 2

Background

2.1 Worked Examples

2.1.1 Definition of a Worked Example

Clark defines a worked example as a step-by-step demonstration of how to perform a task or how to solve a problem" [3]. Another definition for worked examples is given by Atkinson as instructional devices that provide an expert's problem solution for a learner to study. [1]. An effective worked example consists of a problem description, steps towards the solution and instructions at each step representing an experts process of thinking [11]. Of key importance is the step-by-step guidance for reaching the solution. It encourages the learner to form their own explanation for the undertaken step [12] as well as think about what might follow next before they proceed. In essence, worked examples help novices to build an understanding of a concept so that in later stages they will be able to effectively apply this understanding to solve other problems related to this concept.

2.1.2 Worked Examples and Learning

The common assumption that the best learning is by practicing solving problems is not necessarily true for learning Computing Science. Renkl [11] argues that without being exposed to worked examples first, novices have a very restricted knowledge on the domain to be able to effectively reach a solution. Solving problems involves a lot of working memory resources. However, the memory capacity of beginners should be used for building new knowledge instead. Clark argues that solving practice problems leads to using too much memory capacity thus not leaving enough of it for learning new knowledge [4]. Studying worked examples is one of the earliest and probably the best known cognitive load reducing techniques [9]. It has proven to be effective in learning how to solve problems [18]. While worked examples reduce the cognitive load, they also provide a better understanding of the concepts under consideration. This builds up the necessary expertise required to solve a particular type of problem effectively.

2.1.3 Worked Examples in a School Context

The traditional methods of teaching Computing Science in schools across the UK do not include the best proven method to learn a cognitive skill described above. Often in schools, Computing Science concepts are introduced by explaining what the concept is, followed by a simple example. Then pupils are presented with a problem to solve themselves. The jump to problem solving is too quick and the importance of worked examples has

not influenced the teaching methods. Keeping in mind that teachers are often limited time- and money-wise, a possible reason for this is because there is no easy means of finding and adapting existing worked examples to the specific needs of a teacher.

As part of his research project, the former Glasgow University PhD student Yulun Song developed software to facilitate the creation and viewing of worked examples [14]. The thesis statement for the research outlines the basic aims for the project. The system developed is such that it:

- "delivers usable, best practice interactive worked examples to students in a computing science context;"
- "enables teachers to create such interactive worked examples without bespoke programming, and to evolve them on the basis of feedback from the students."

Song was particularly interested in Computing Science problems due to their transformation-based nature. They involve the analysis and the transformation of one representation of the problem, such as text definition or a diagram, into another representation, i.e. the solution. An example described in the research thesis is building a database system from a specific set of requirements expressed in the form of a problem description in human language. The text describing the problem needs to be transformed into a graphical representation of the same problem - an ER diagram, which is then translated into a machine language such as SQL. Judgement and decision-making play a huge role in solving such a problem. However, these only come with experience and in order to gain such experience Reed and Bolstad [10] claim that one example- which is the typical case in schools- might be insufficient. In his research, Song argues that a system that enables the user to view multiple worked examples would prove efficient in such a context. He therefore developed a tool for the provision of worked examples in Computing Science so that the user is exposed to more than one of those examples.

2.1.4 Problems with Existing Methods for Delivering Worked Examples

One can argue that worked examples can be found in many books and lectures so at first it may seem questionable what value would software bring to the existing provision. Song's thesis [14], however, raises some strong arguments to be taken into consideration, as shown below.

- The worked examples in books or lectures are not interactive enough. The readers of books or the attendees of a lecture are presented with some examples, but often the process of thinking why a particular action is undertaken or is a better option for reaching a solution remains unexplained. One can argue that the university context has some grounds for interactivity or discussions. Yet many students may not exploit this due to shyness or simply because they might not know what questions to ask. Even if some interaction happens, this is not recorded or captured as part of the teaching process so the students cannot go back and review it.
- 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 - and Song aims to address - is to adjust the worked examples depending on the teaching or learning needed.
- Worked examples in books provide little or no feedback on how they were used to the author or to teachers who benefit from using such examples in their teaching. For example, the only available information for a book would be the number of copies sold. This would not provide any insight on the value the examples brought to the reader. What is desired is information about how a particular worked example was used, were there any problematic areas and how the readers benefited from it. Such information would allow

the authors to improve their future work at constructing worked examples. In addition, this information could be beneficial to teachers or lecturers who could use it for assessing what parts of the example were problematic and adapting their teaching accordingly.

2.1.5 How Does a Computer-Based Application Solve the Problems with the Traditional Methods of Delivering Worked Examples?

The piece of software proposed and developed as part of Song's work aims to address all the issues mentioned above. The student becomes actively involved with the material since revealing the steps required to reach a solution is under their control. The entire problem solving process can be fully captured and the students can easily go and review parts causing confusion. Complete explanation of every step is provided, enabling the student to follow the process of thinking of an expert. Revealing the solution step by step encourages thinking about the next logical step and guides the student towards the correct direction of thinking before they get confused. In addition, usage data can easily be captured to give feedback on how these worked examples were used. Data intended to be collected includes time spent at each step and answers to any questions present in the examples. Such information can be beneficial to two groups of people:

- Authors of worked examples. They could benefit from the knowledge of the time the majority of the students spend at each step. If this time exceeds dramatically the expected time for this step, this can be an indication that the step is unclear and brings confusion. Knowing this, authors could update the example by including a better explanation or by breaking this step into more than one steps and examine the effect this has. This way authors will learn how to build their examples and this will also bring benefits to the reader in terms of provision of improved worked examples.
- Teachers. The knowledge that their students visit a step multiple times or spend too long before proceeding would indicate to the teacher that their students do not understand the material for this step well enough and they might need to revisit it in class.

2.2 Interactive Worked Examples Tool

The following subsections provide more information about the IWE tool.

2.2.1 Intended Flow of Interaction

There are two well-distinguished groups of users of the IWE tool authors of examples, and students. Each group is serviced by a separate interface of the application. The flow of interaction of these groups with the system, as presented in Song's thesis, is shown on Figure 2.1. The original figure can be found in Song's thesis as Figure 2.6 (p.57). The following bracketed numbers correspond to the associated numbers in the Figure.

- Authors can create (1) and modify (5) interactive worked examples through the author interface. This interface also presents them with any student feedback (4) on these examples.
- The student interface serves as a worked examples viewer (2) where students are given the opportunity to ask questions and write comments (3).
- The interaction between teachers and students is direct rather than through the system.

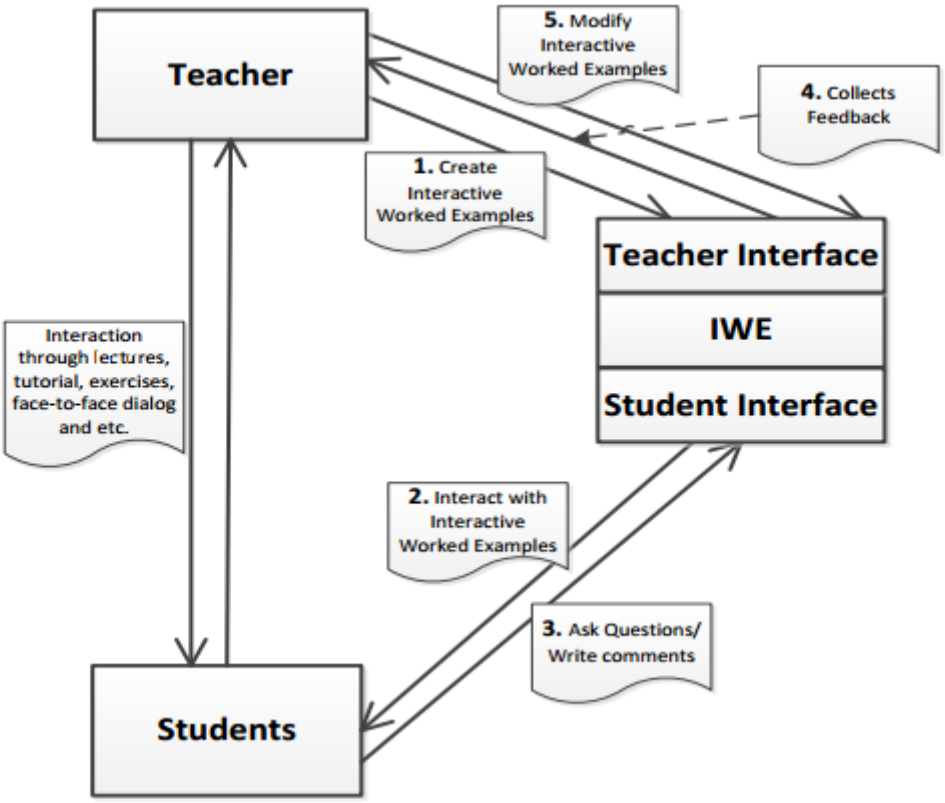


Figure 2.1: IWE Flow of Interaction.

2.2.2 Structure of IWE

IWE stores the worked examples in XML files. There are three types of files that are of particular interest for this Level 4 project- `Documents.xml`, `Applications.xml`, and `Processes.xml`. The structure of those files is graphically represented on Figure 2.2. A description of each file can be found below.

- The `Documents.xml` file stores the collection of documents created by an author. A document is one of the representations involved in a particular worked example- perhaps it is the problem specification, or an intermediate solution, or the solution. It is split into fragments which are small logically separated portions of the document. The reason for splitting the document into fragments is so that the document can be revealed gradually, to show the step-by-step problem solving process. Individual fragments can also be highlighted to be brought to the viewers attention.
- The `Applications.xml` file stores layout information about worked examples, bringing together the particular documents involved in the worked example. The way the documents are laid out visually, in panels, is recorded.
- The `Processes.xml` file defines the steps for the worked examples. For each step there are a number of changes and an explanation of those changes. There are two types of changes. The first type specifies which fragment of a document is involved in this change. These fragments can be shown, hidden, highlighted or unhighlighted depending on the effect the author is aiming to achieve. The second type of changes corresponds to a question and possibly a set of options the user can select from in an attempt to answer it.

There is one more type of XML file which is not shown on Figure 2.2. It contains information about different styles that can be used for the worked examples- there is a similar, although simpler, version of the style mecha-

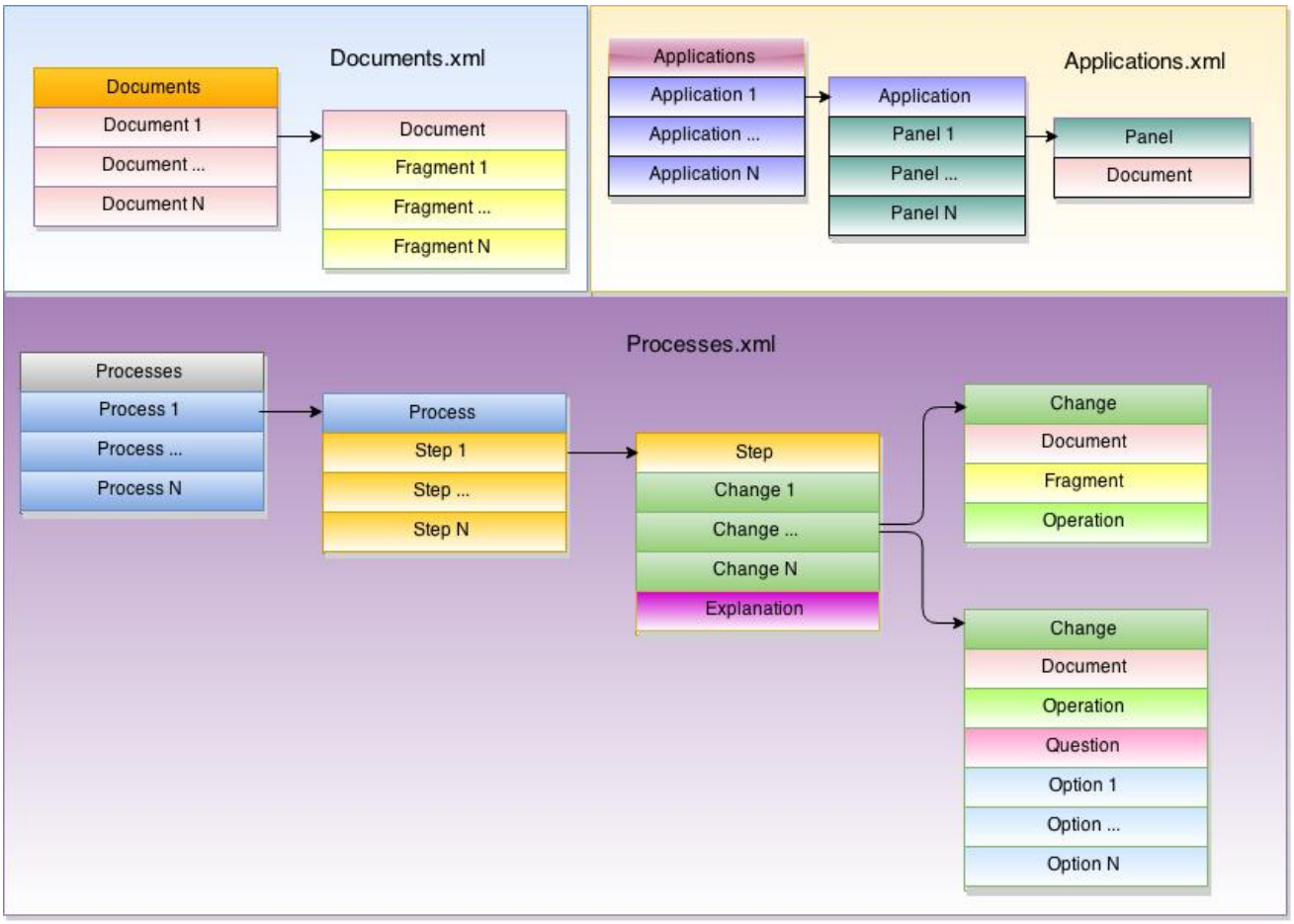


Figure 2.2: Structure of the XML files storing the worked examples created in IWE.

nism found, for example, in word processors or CSS style sheets. There is no need for this file to be discussed in any detail. However, the reader needs to know that documents have styles associated with them depending on the type of document, enabling different fragments within a document to be shown in different typographical styles.

2.2.3 Main Features of the IWE Student Interface

The student interface aims to provide an effective worked examples viewer. A screenshot of the final version of Song's prototype can be seen on Figure 2.3. The most important characteristics are labeled with numbers and are detailed below.

1. An area for showing the worked examples installed on the system and enabling the user to choose an example to work on.
2. Panels showing different documents for a particular worked example.
3. An area for controlling transitions between steps.
4. An explanation area where the expert's process of thinking involved on the current step is shown.
5. Highlighting of the newly appeared text at a particular step for drawing the users attention to the new content relevant for the current step.
6. Highlighting of fragments of interest for a particular step.

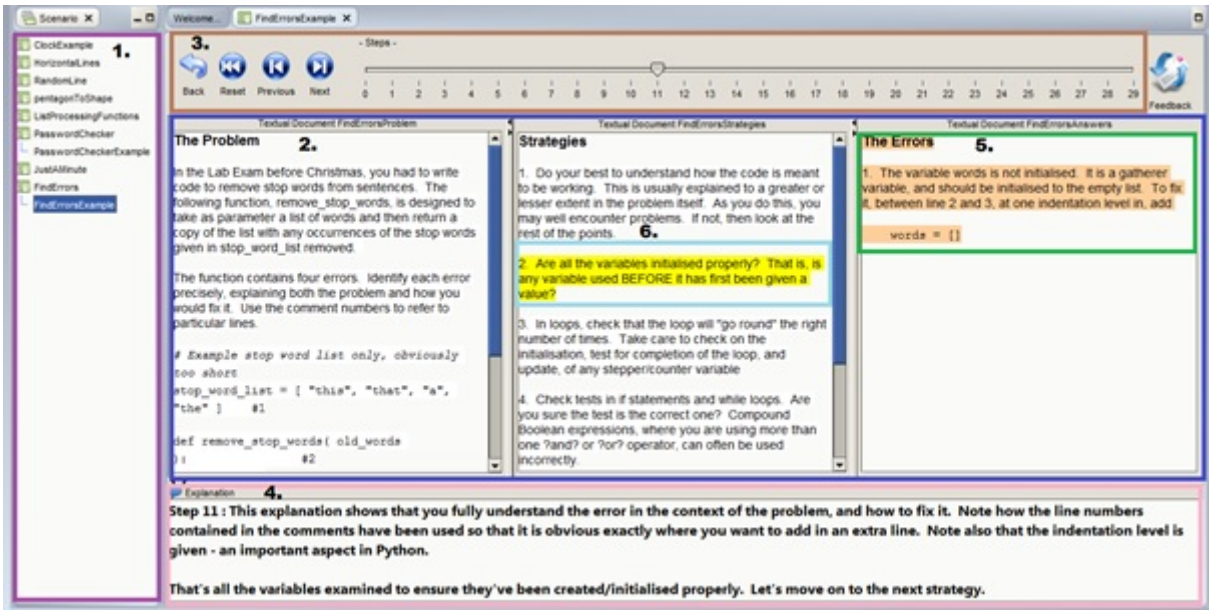


Figure 2.3: A screenshot of the student interface of IWE.

As the student uses the controls in area (3) 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. Other features of the IWE 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.

2.2.4 Relation of IWE to this Level 4 Project

The evaluation on IWE conducted as part of Song's research clearly shows the benefits of the tool. It has proven to be well-accepted and valuable as a technique to enhance a students learning experience. It also achieved its goal to enable teachers to more easily and quickly develop worked examples to fit their needs. The aim of this prototype, however, doesn't cover deployment of the software in educational institutes but rather it has demonstrated that it would bring benefit to both students and teachers.

The motivation for this Level 4 project is to make use of the findings in Song's research and take a step forward in deploying the software in schools across the UK. The focus of the project will be reconstructing the student interface of IWE into a web-based form, as well as providing a new interface for teachers to separate them as a distinct group of users. The author interface of IWE is beyond the scope of this project and no understanding of any of its aspects is needed by the reader to follow this dissertation.

Chapter 3

Requirements

This chapter provides a detailed description of the functional and non-functional requirements for WEAVE.

3.1 Preface

As described in Chapter 2, this Level 4 project builds upon an existing system for facilitating the use of worked examples in educational context. The evaluation of IWE clearly shows that such software would be a valuable asset contributing to the learning process of students. Due to the overly complicated procedure required to deploy IWE in schools while it is in the form of a Java standalone application, the need to turn it into a more easily deployable online version arose. At a later stage of the project, interviews with highly motivated teachers, who are part of the *PLAN C Project*[1], have identified the need for one more interface to be used in schools. In order to improve their teaching practice and to be able to provide high quality feedback to their pupils, these teachers would benefit from knowing how pupils in their classes use these worked examples. Information that would be valuable for them includes:

- identification of which pupils interacted with which examples;
- aggregated information on answers selected for multiple choice questions and the pupils that selected each answer;
- answers to any open-ended questions;
- information about the average time spent at each step of an example as per the whole class (or an individual); and
- summary data of the total time spent at an example and the last step reached by each pupil in the class.

The requirements for the system developed during this project are presented in a single set. However, they emerged in two phases as the project progressed. For the reader to understand some of the results presented later in the dissertation, the timeline of activities is first presented.

September Requirements for web-based viewer drawn up with client Quintin.

October-November Reverse engineering of IWE to understand data format, and development of web-based viewer interacting with IWE

[1] `http://www.cas.scot/plan-c/`. Last accessed March 24th, 2015.

December-January Finishing implementation of viewer, and planning for evaluation began. Consideration given to anonymisation of data captured for us, while keeping it identifiable for teachers. Realisation of need for a series of interfaces - author/experimenter - teacher - pupil. Requirements capture from teacher (Peter)

February Implementation of teacher interface

March Evaluation and write-up.

Combining the two sets of requirements, the project aims to achieve four goals.

G1. 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.

G2. Provide an interface for teachers that will help them gain more information on how the worked examples are used by their own pupils.

G3. Replicate as closely as possible the student interface of IWE.

G4. Ensure that worked example authors can view usage data in an anonymous manner, such that individual pupils, classes or schools are not identifiable.

3.2 Classification of Requirements

The requirements are classified according to the *MoSCoW* [7] classification method. The categories considered are:

- **must-have**- requirements that are crucial for the achievement of the goal of this project and must be implemented.
- **should-have**- requirements that are considered to be important but not necessarily crucial for achieving the goal of this project and should be implemented.
- **could have**- requirements that have been identified as features that would add further value to the prototype but are thought of as stand-out ones rather than ones contributing to the correct functioning of the prototype and may not be implemented due to constraints.

The **would-like** category coming from the **W** in *MoSCoW* is not part of the classification methods used for this project due to the fact that all the requirements fit comfortably in the other categories.

3.3 Functional Requirements

3.3.1 Interoperability with the Existing Author Interface

The existing system uses XML files to store the worked examples. The structure of these files is shown on Figure 2.2 in Chapter 2. The web-based system will need to read in worked examples from these data files. Furthermore, feedback from pupils and teachers will inevitably lead to changes being required in some of the worked examples. The existing authoring tool supports editing of worked examples, and it is expected that it will still be used to make such changes. WEAVE will need to be able to support these changes. Due to the fact that

the update model of IWE is destructive- no versioning of the examples is supported- and that WEAVE does not provide means for modifying examples, the update model will need to follow the one of the old system.

The prototype:

- **must** be able to parse an XML file containing the fragmented problem specifications of the worked examples and their solutions.
- **must** be able to parse an XML document containing information about individual steps of the worked examples (e.g. which fragments of a document must be shown/hidden/highlighted/unhighlighted, the explanation associated with a step and a question if one was provided).
- **must** be able to parse an XML document containing information about the layout of worked examples (e.g. number of panels for the example, their order and problem solutions associated with each panel).
- **must** be able to parse an XML document containing information about the styling associated with each example (e.g. font style, font size, etc.).
- **must** be able to support easy addition of new worked examples created using the old authoring tool.
- **must** be able to incorporate new versions of worked examples already installed in the web-based system.

3.3.2 Teacher Interface Requirements

A major part of the contribution of WEAVE is to enable teachers to receive information about how their pupils worked with examples, while authors of such examples and Computing Science researchers must receive such data in an anonymised way. The desired effect is for teachers to be able to see usage data for their classes as well as individuals in these classes. However, protecting the privacy of both teachers and pupils is a major issue. The authors of worked examples will be able to see any usage data for the examples they created. If this data is informative enough for them to identify the person standing behind this data, this would be highly unethical and would violate privacy.

In this section, the requirements for the teacher interface are outlined. The next chapter will describe how the privacy issues mentioned above are resolved by the system and will discuss in further detail these requirements.

The teacher:

- **must** be able to choose whether to track their pupils' progress or not.
- **must** be able to view how the students in a particular class interacted with the examples.
- **must** be able to view how a particular student interacted with the examples.

3.3.3 Replication of the IWE Student Interface

One of the goals for this Level 4 project (G3) is to replicate the student interface of IWE as closely as possible. The reasoning behind the requirements for the student interface, as well as the positive conclusion from their evaluation, are described in detail in Song's thesis. These were found acceptable for the purpose of this project. In short, the research literature describes the qualities of successful worked examples [11], and these qualities were included in Song's design. Their implementation in his software system was shown to be suitable for use by students in the extensive evaluation carried out.

The prototype:

- **must** enable the pupil to select a worked example from a list of existing examples.
- **must** support multiple panels for the different parts of the problem solution.
- **must** contain a dedicated area for the explanation.
- **must** support showing/hiding/highlighting/unhighlighting of fragments.
- **must** support the option to ask pupils questions.
- **must** enable the pupil to go back and forwards through steps.
- **must** highlight the newly introduced fragments at each step.
- **should** record time spent at a step.
- **should** record answers to questions.
- **should** enable the pupil to reset the example there are working on.
- **should** provide a means for drawing the pupil's attention to the newly introduced fragments.

3.3.4 Additional Features Needed for the Student Interface

This section describes the requirements for satisfying goals G2 and G4 to support identifiable usage information for the teacher, and anonymous usage information for worked examples authors and for Computing Science education researchers. It is important that each teacher is able to link usage data to their groups/pupils, while authors of such examples must not be able to identify by any means what the group or who the pupil is due to the privacy issues discussed previously.

The prototype:

- **must** be able to record usage data for an example in such a way that the pupil's teacher will be able to know who the data belongs to.
- **must** be able to record the usage data for an example in such a way that the true identity of the pupil associated with this data is not revealed when this information is sent to authors of worked examples and Computing Science education researchers.
- **must** allow the pupil to use the system without any identifying information.

3.4 Non-Functional Requirements

The non-functional requirements for WEAVE are guided mostly by the web-based nature of the system and by the context it is intended for. In order for pupils to be able to study the worked examples effectively, and also due to the small workstation screen sizes found in schools, the area showing the worked example should be maximised. Furthermore, due to the step-by-step nature of worked examples, some steps may put more emphasis on the explanation while others might be more intensive in the problem specification areas so the system must be able to deal with such situations accordingly. In addition, features which make the interactions with the examples

more convenient and which would minimise effort, such as shortcuts and appropriate fitting of the whole system on the screen, are highly desirable.

The web-based nature of WEAVE poses a possible problem when uploading modifications to existing examples because pupils might be working on these examples at the same time. Consistency must be ensured in such cases, meaning that the pupil should be able to see either the old or the new version of the example, rather than a mixture of both.

Since pupils may have not worked with such a system before, they may benefit from a brief guide on how to use WEAVE in an optimal way.

These considerations form the following requirements:

- The prototype **must** be easy to use.
- The worked examples **must** fit the entire screen.
- The size of the area showing the worked examples **must** not exceed the size of the screen.
- A modification to a worked example **must** not affect pupils doing the same example.
- The student interface **should** include a tutorial on how to use the system.
- The teacher interface **should** provide information on how to use each feature.
- The panels showing the problem content **should** be resizable.
- The explanation area **should** be resizable.
- Shortcuts for easier transition between steps **could** be added.

Chapter 4

Design

This chapter outlines the design decisions for this Level 4 project. As it was previously described, WEAVE is based on the earlier Java standalone application IWE. This affects to a great extent these decisions. Following the well-established software engineering principle of reusability, guidance for the design of WEAVE was to reuse any good aspects of IWE's design while improving its weaknesses.

4.1 Storage of Data

The scope of this Level 4 project does not include the creation of a web-based author interface. This means that the design of WEAVE must ensure an easy and efficient way for importing and storing the existing worked examples. In addition, WEAVE must support uploading updates to existing examples. As described in Chapter 2, the examples are stored in the form of XML files. To remind yourself of the structure of these files, please refer to Figure 2.2.

During the exploration process of IWE, a weakness of storing the information about the examples in the XML files used by IWE was identified. Since these files are easy to access and modify by the authors of worked examples, and this is typically much easier than modifying the worked examples using the author interface, one may be tempted to make changes to the examples manually. However, references to some objects may be present in more than one of these XML files. For example, in the `Documents.xml` file, fragments are stored as individual elements identified by a fragment id. When the steps for the example are defined in the `Processes.xml` file, each fragment is referred to with its id and also, for an unexplained reason, the fragment text is included. A problem with storing information about the examples in files is that if an element is modified in one of the files, consistency about this element must be ensured across all copies, or else the system fails. It is not trivial having to find the same feature across multiple files and in the end consistency and validity are not guaranteed. This appears to be a case of poor software engineering, as there is no apparent reason for the fragment text to be copied between XML files, and not just the fragment id.

In the new system the worked examples are stored in a database where the relationships between different objects are expressed via foreign keys. So, for example, the same fragment is not copied but referred to by its foreign key. In addition to ensuring consistency, using a database adds a level of reliability that the data is valid because internal integrity checks are made before saving an object to the database. Further advantage is that the Django administrator interface allows an easy means of exploring and modifying the examples in a consistent and safe manner.

4.2 The Data Model of Worked Examples

The way the worked examples are structured in the XML files generated by the author interface of IWE is shown in detail on Figure 2.2. The bullet points below can be used as a quick reminder. The structure of the XML files determines to a great extent how the worked examples are stored in the database. Figure 4.1 shows the UML Entity-Relationship diagram for the worked examples in WEAVE. The numbers associated with a particular relationship between the entities correspond to the numbers in the bullet points below to illustrate the similarities between the two methods of representation of the worked examples for the different systems.

- Each worked example consists of one or more panels (1), which show the text of a document for the worked example (2).
- The text of each document of a worked example is split into different fragments (3) (portions of the text for the example).
- Processes define the steps for each example (X). There are many steps for each worked example (4).
- Each step of an example consists of a set of changes (5) and an explanation (6).
- A change can involve either showing/hiding/highlighting/unhighlighting a fragment (7) of a document (8) or asking a question (9) (open-ended or multiple choice (10))

Please note the bullet point that has X instead of a number. The `process` element in the XML files was found unnecessary when storing the worked examples in the database and it would have led to redundancy. Instead, each worked example (referred to as an `application` in both Figure 2.2 and Figure 4.1) is associated with a set of steps. The order of these steps is determined by the `order` attribute of the Step entity. When a worked example is loaded, all the steps for this example are retrieved and are shown in the order defined by their `order` attribute.

Please note that this is a simplified version of the objects storing the worked examples for illustration purposes. As with the XML files, there are objects for the different type and style of documents and fragments. Inserting them in the UML Entity-Relational diagram on Figure 4.1 would have made it unclear and difficult to read. The type and style of documents and fragments are defined using foreign key relationships with the respective objects. On Figure 4.1. these foreign key relationships are expressed by adding *(FK)* next to the name of the object for the particular type or style involved in this relationship.

4.3 Design for the Translation of the Worked Examples from XML Elements into Database Objects

The database is populated with the elements stored in the XML files produced by the author interface of IWE via a command line interface. A population script- `population_script.py` file- is run on the command line. It takes one optional parameter to specify the path to the XML files. If this parameter in not provided, the script will look for the files in a default directory for storing the examples. More details on the implementation of this population script are provided in Section 5.2.2. Instructions on how to run the script and where the example directory is are provided in a `readme.txt` file in the project.

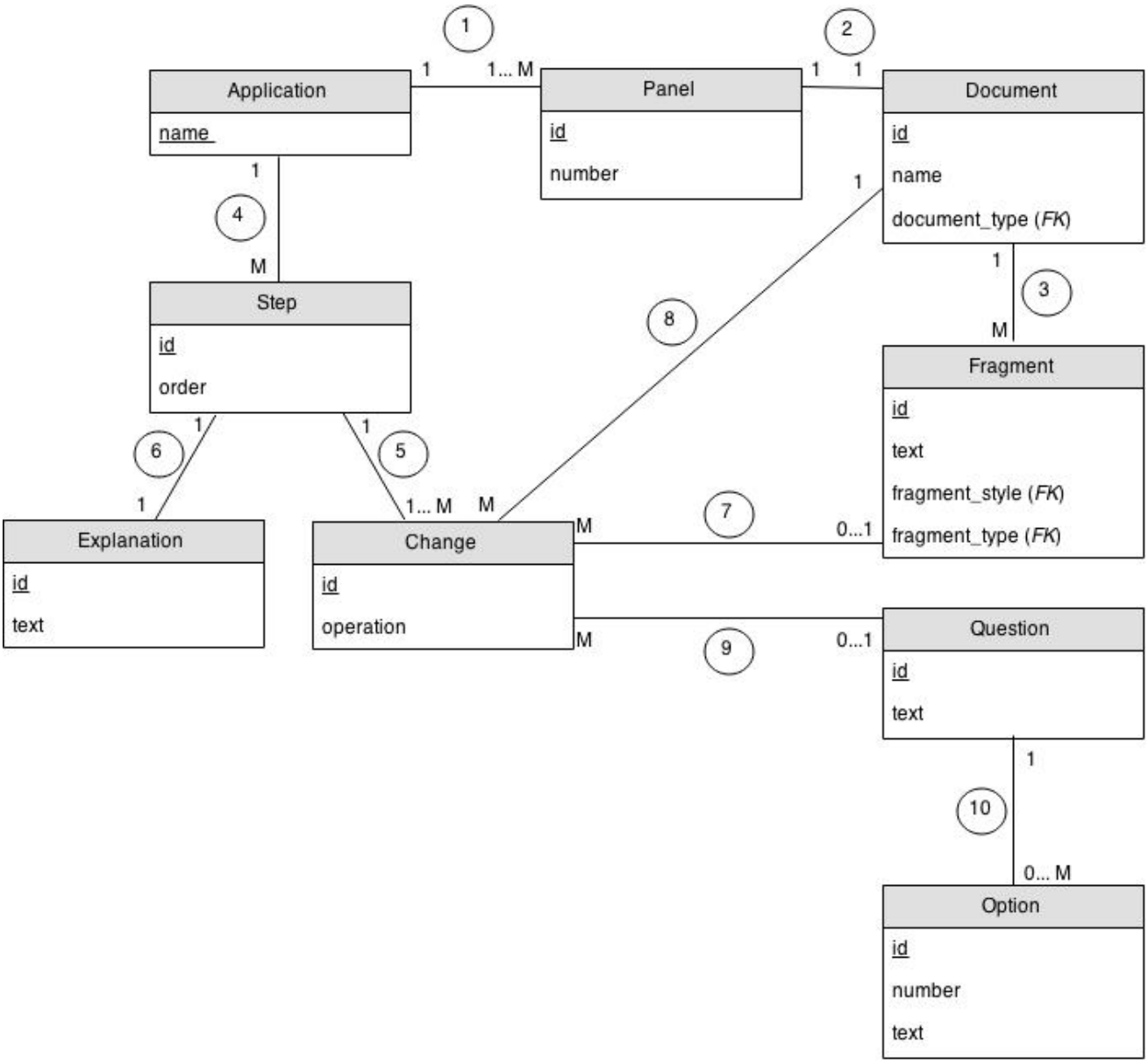


Figure 4.1: A UML Entity-Relationship diagram for the worked examples.

4.4 Authentication and Privacy

A significant discussion point was how to authenticate teachers and their students due to the privacy and ethical issues discussed in Section 3.3.2. As a reminder, the main issue around privacy/ethics is to avoid storing information in the system that could identify individual schools, teachers or pupils. If this can be achieved in the design, then data gathering for worked example authors and system designers, essential to improve both worked examples and the system itself, can be carried out with no requirement for informed consent for data usage by the pupils or teachers. The requirement here, then, is that names for teachers, classes and pupils do not contain identifying information. One possibility was for students to create their own accounts and give their usernames to the teacher, so that he/she could monitor their progress. However, this approach could potentially result in various complications.

- Pupils would need to be explicitly instructed that their accounts should not reveal their true identity, since this information would be stored in the system database, and visible to systems administrators.

- Pupils would not have any benefit of having their own accounts as only the teacher will be the one who would use their usernames for something meaningful, i.e. to check their progress.
- This approach solves identification at an individual level but each pupil needs to belong to a group as well, to assist the teacher in monitoring the progress of a particular class, so this becomes an additional management overhead for the teacher.

A second option was considered, in which the teacher would create an account for each of their classes. For this account, they would need to create usernames for their pupils. Keeping in mind that a teacher would often have more than one class and that each class consists of twenty to thirty pupils, the following problems arise:

- The teacher would need to create a lot of accounts and this could be a non trivial and time consuming task.
- The teacher would need to ensure that they will be able to match each of their pupils to their id since they must not use any names due to the privacy issues mentioned above.
- Account names should not be guessable by other pupils, to avoid any one of them pulling a prank on another pupil by using their username. Hence the teacher can't use an easy sequence of anonymous usernames such as `p1`, `p2`, `p3`, etc. On the other hand, long, anonymous usernames should be avoided since these are tedious to store and transmit and are likely to lead to errors in use.

To get around the privacy issues and the problems with the options described above, a third approach based on the general idea of the second one was adopted. The main points of this approach are:

- Teachers create their own accounts, with a username that does not identify them or their school.
- Once logged in, they will be able to create groups, one for each of their classes, with a name that does not identify the school.
- On creation of the group, the teacher needs to specify the number of the pupils in this class. WEAVE will then generate random ids for these pupils. An id consists of two random letters followed by a single digit. While being short enough to be easily remembered, such an id ensures that the privacy of pupils is protected due to its random nature. Furthermore, the number of possible combinations for all pupil ids is large enough so that it will be highly unlikely that pupils will be able to guess one of their classmates pupil id and work with the examples on their behalf.

Talking to a lead teacher, Mr Peter Donaldson, who is part of the PLAN C Project, a potential inconvenience of this approach was identified. Using WEAVE for longer periods than one academic year could lead to a significant increase of the groups. In addition to the growth of the number of groups, some teachers might prefer to use the same name for their classes across years. To resolve these issues, a further classification of groups by the academic year the group belongs to was adopted.

Mr. Donaldson was generally happy with the idea that teachers select the number of pupils for each group at the creation of this group. However, he pointed out that it is possible for a pupil to arrive in a class at a later stage than the beginning of the academic year. Using the selected approach would have required that the teacher creates a new group just to add one pupil only. This could be very problematic, because data about the same pupils would be spread across two different groups and most of the pupils will be given two pupil ids which could become really confusing for both teachers and pupils. To avoid these problems, the option for teachers to update a group was added to the design decisions for the authentication part. In addition, groups can also be deleted in case of creation of unneeded groups.

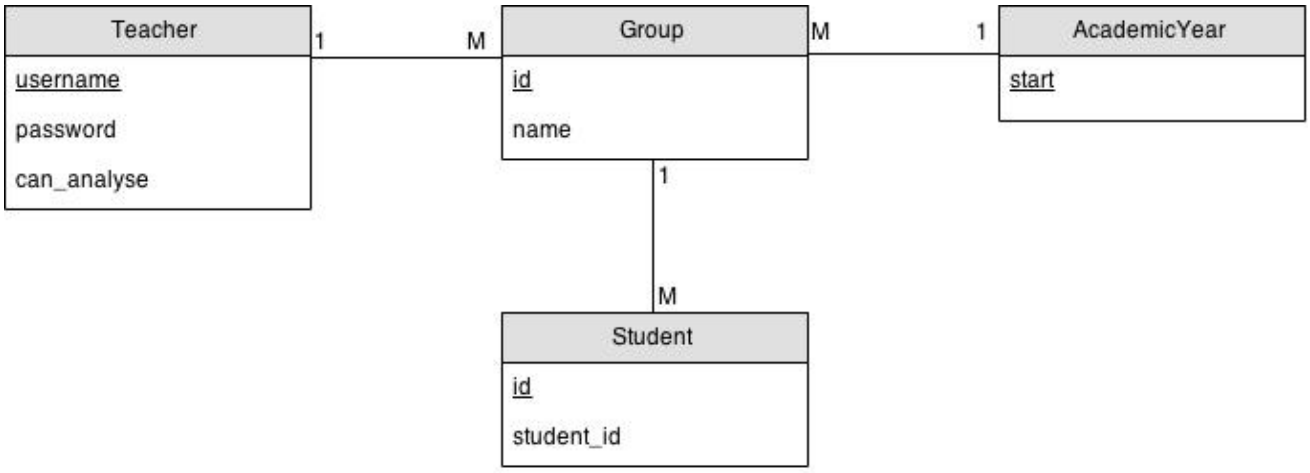


Figure 4.2: A UML Entity-Relationship diagram for the users of WEAVE.

4.5 The Data Model for the Users of WEAVE

There are four classes in the database which represent the special relationship between pupils and their teachers and classes. These are shown in the UML Entity-Relationship Diagram shown on Figure 4.2.

Each teacher can have many groups. A group is associated with a particular academic year. This would allow the presence of many groups with the same name for a particular teacher, as long as they are in different academic years. There can be many groups for each academic year. Each student belongs to one group. There are many students for a group.

4.6 Presentation of Data to the Teacher

The main goal of the teacher interface is to present to the teacher data associated with a particular group or pupil. Three different types of data are recorded from the student interface:

- Time at each step.
- The direction of the transition to each step i.e. is the pupil going backwards or forwards to a step.
- Answers to questions.

Careful consideration was needed to reach a solution that would visualise this data in a way which would enable the teacher to comprehend it easily and encourage further analysis. In addition, the ability for the teacher to be able to view data both at a class and at an individual level further influenced the design decisions.

In a discussion with Mr. Donaldson- as a teacher who would use such data- it has been decided that presenting the data in the form of graphs would be beneficial to teachers as they would be able to identify patterns and any exceptional events for a particular worked example. These graphs should reveal information about the performance of the whole class as well as of individual pupils at each step of the example keeping in mind that some steps have questions. For this purpose, five different types of graphs were decided upon:

- **Average Time.** This graph would show the average performance of the pupils in the whole class on a particular example. Different steps will be represented on the x-axis of the graph by their step number. The

average time spent on each step will be shown by the y-axis. Since the x-axis consists of the step number, this is not very informative to the teacher because they would need to look at the actual example to remind themselves about the context of the step. To avoid the need for that, hovering over the point representing the step will show the beginning of the explanation. Clicking on that point will open a window showing the whole text of the explanation, the average time spent on that step and how many pupils made a backwards transition to the step. As mentioned above, some steps contain a question, rather than an explanation. Such steps will be identified by a question mark in front of the step number on the axis label. Instead of showing an explanation on mouse hovering, a message encouraging teachers to click in order to see pupils' answers is shown. Clicking on the point for that step will show the question, and a bar chart with all the possible answers and how many and which pupils selected each option.

- **Student Time.** This graph is conceptually the same as the average time graph. Instead of showing information about the whole class, however, it shows the total amount of time spent at each step by a selected pupil.
- **Student Answers.** This column chart shows the options for a selected question and the number of pupils who chose each option. Hovering with the mouse over each bar shows the list of pupils who selected the answer represented by this column.
- **Class Steps.** This graph shows information about the time spent at a chosen step of an example by the pupils of a class. It is in the form of a column chart where each column represents a pupil's attempt. This means that there might be more than one column for each pupil if they have attempted the selected step more than once- each column revealing information about the time spent by that pupil at a particular attempt of the step.
- **Class Summary.** This is a table showing summary information about the total time spent by each pupil at a particular example, how many times they returned to previous steps and the last step they reached. This would show the teacher how much effort did the pupil put in each example, how many problematic or unclear steps they encountered as well as whether they completed the example or which step they gave up on. This is ideal as a quick overview of a class's progress, highlighting problematic pupils.

4.7 Architecture

The architecture of WEAVE consists of three distinct tiers as visualised on Figure 4.3.

- **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 HTTP GET or POST requests.
- **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. In this tier the requests from the client are parsed and translated into *ORM (Object Relational Model)* 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 an ORM response, the middleware is responsible for translating it into an HTTP response and passing it back to the presentation tier.
  - The second component is the connection point between IWE and WEAVE. This is where the translation of the XML elements takes place, so that the examples can be stored in the form of ORM objects. This translation is done with an XML parser. The relationships between these elements are established and the database is populated with the objects defined by the parser.

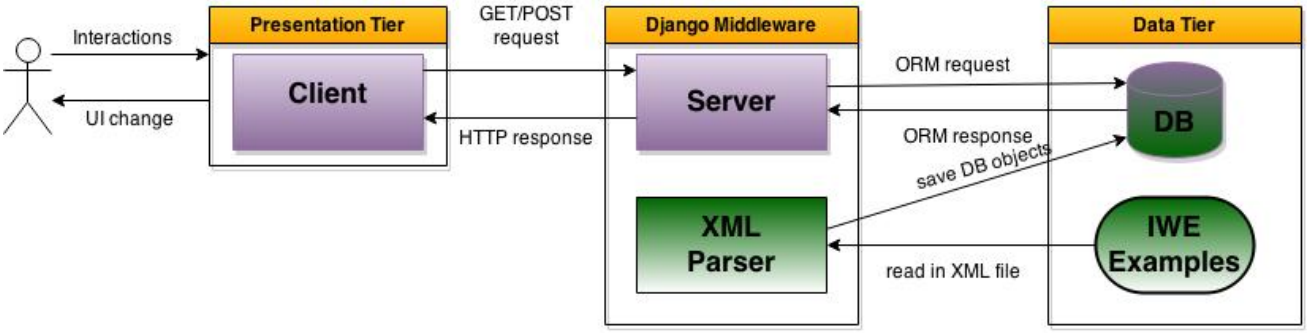


Figure 4.3: The N-tier architecture of WEAVE.

- **Data Tier.** This tier represents the database in which all the information used or generated by WEAVE is stored in the form of database objects (typically known as rows in a relational database). 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.

4.8 User Interface

As already mentioned, WEAVE is intended for three distinct groups of users, each with their different needs. However, for the purpose of this Level 4 project, only two of these groups will influence the user interface. Addressing the needs of authors is beyond the scope of this project. This is an intermediate step between moving from using entirely IWE to using entirely WEAVE. For this purpose, no author graphical user interface is provided. To create new examples, authors need to use the author interface of IWE. These examples are then added to WEAVE via a command line interface as pointed out in Section 4.3. This is why the section on the user interface is split into two subsections only, which describe the user interfaces for pupils and for teachers.

4.8.1 Student User Interface

A core purpose of this Level 4 project is to translate the student part of IWE into a more easily deployable online version. Careful consideration about the layout of the student interface of IWE is evident. The evaluation of IWE proved that the current interface is well accepted by students. A screenshot of this interface is presented on Figure 2.3. Generally, it has been decided to take advantage of Dr. Songs findings and to reuse a very similar interface.

4.8.1.1 Home Page

The need for some additional features of this interface arises due to the fact that teachers need to be able to identify their pupils in order to monitor their interaction with the examples. To ensure that pupils are using the system in the intended way and to encourage them to use the details provided by their teacher, the examples will be hidden to them until they log in either by entering some authentication information or by identifying themselves as anonymous users.

Once the pupil has authenticated with the system, the area prompting the user for details is exchanged for the list of worked examples to encourage them to choose one to work on. Due to the fact that the system is required to

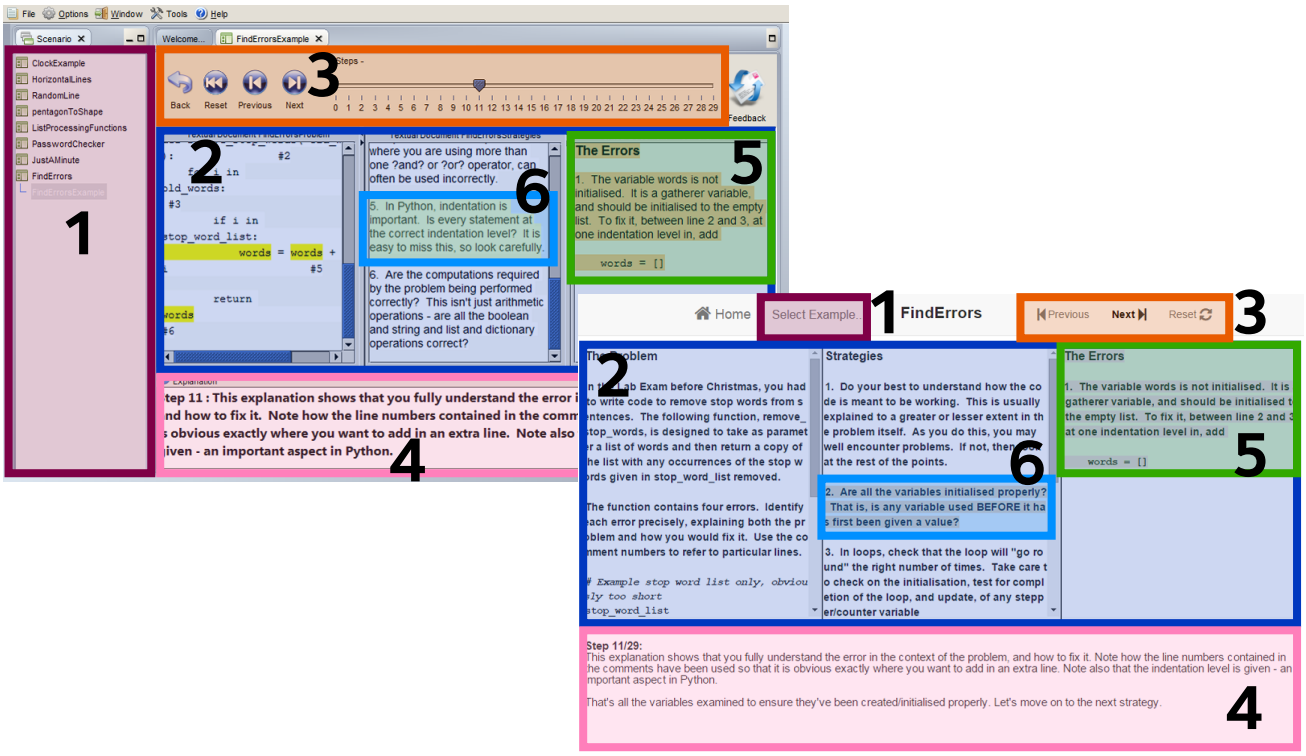


Figure 4.4: Screenshots from the student interface of IWE (left) and WEAVE (right.

accommodate examples created by many teachers across the UK, a filtering by the name of the worked examples functionality has been provided.

In order to be able to exploit the worked examples viewer in an optimal way and to familiarise students with it, a tutorial appears on the main page. The idea for having a tutorial was borrowed from IWE. However, the way the tutorial was constructed there was identified as potentially ineffective at communicating all the information the pupil needs to know before working on examples due to the fact that it contains a lot of text which may discourage some of the pupils to read it. Furthermore, even if they read the tutorial, they may not understand what is referred to in the text because they may have not seen the worked examples viewer and its features in advance. A different approach was chosen for the tutorial of WEAVE. It is split into different steps describing an individual feature using minimal text and a screenshot of the feature. Screenshots of the tutorials in the two systems are shown on Figure 4.4. for comparison.

4.8.1.2 Page for Viewing an Example

The page for viewing a worked example is very similar to the one used for the IWE student interface. Figure 4.5 shows screenshots of the student interface for each of the systems to illustrate the changes to the interface discussed below. Due to constraints imposed by the size of the screens in schools, the design of WEAVE needed to be adjusted accordingly. The area for selecting an example (referred to as **Element 1**) is placed on the navigation bar with all the examples appearing in a drop down menu on request. This saves a significant portion of the screen which can be used for the problem specification instead. Another space consuming element is the bar showing the current step (**Element 3** in the IWE student interface). In WEAVE this information is shown as a part of the explanation (**Element 4**) instead. The elements surrounded with boxes with the same colour and having the same number have an equivalent functionality across the two systems. The reader can remind themselves of the purpose of each area by referring back to Section 2.3 in Chapter 2. A new feature to ensure drawing the pupils' attention to the newly appeared content of the example- automatic scrolling to this content- has been added to

the student interface of WEAVE.

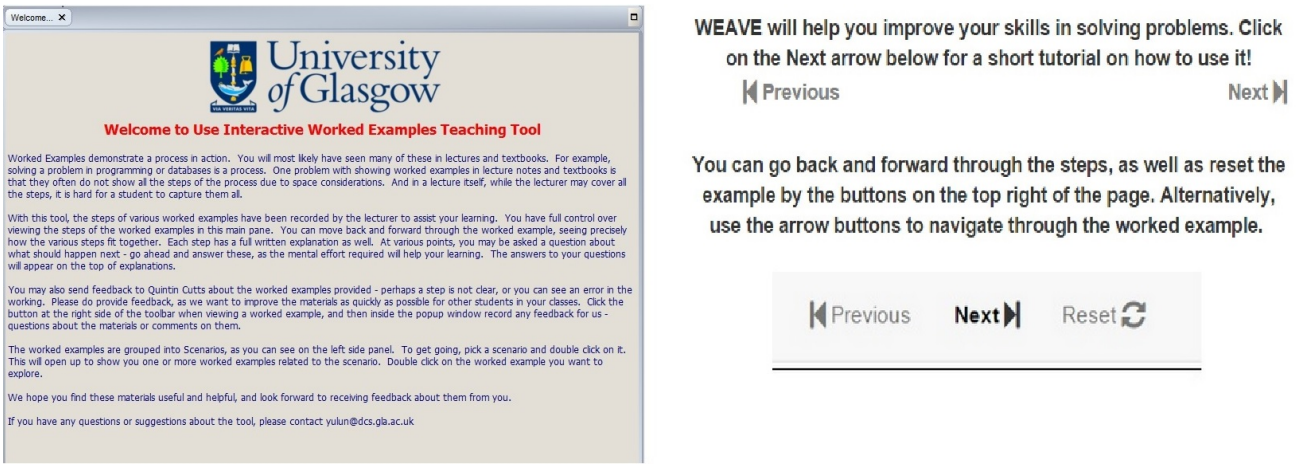


Figure 4.5: Screenshots from the tutorial of IWE (left) and WEAVE (right).

4.8.2 Teacher Interface

4.8.2.1 Non-logged in Teachers

The purpose of the teacher interface requires the teacher to be logged in. Therefore, on the first visit of the page the teachers are presented with registration and log in areas only.

4.8.2.2 Logged in Teachers

The discussion in the design decisions on the authentication to WEAVE (Section 4.4) and the way usage data is presented to the teachers (Section 4.6) identifies the main features of the interface for the logged in teacher. Options for all the activities a teacher can undertake via the teacher interface are present on their home page to avoid the need for transitions between different pages and to simplify navigation of the website. The metaphor here is of a control dashboard. The wireframe for the interface can be seen on Figure 4.6.

The main page is split into three areas.

- **Area for registering, updating and deleting a group.** These three options are provided in the same area on the screen. When the teacher selects the desired option, the elements for this area change accordingly. For example, when the user wants to create a group, they need to enter the group name and the number of students for that group. On update or deletion of a group, on the other hand, they select the group name from a dropdown list. The list of existing groups is shown to remind which group names are unavailable to this teacher. The textbox for entering the number of students accepts integer input only for error prevention purposes. On the submission of the request to create/update/delete a group, a message confirming the status of the action is show.
- **View Group area** enabling the teacher to select a group for which to view the student ids. Again, for simplicity and error prevention, the teacher selects the group via a dropdown list rather than typing its name.
- **View Progress icon** which navigates the teacher to the statistics page.

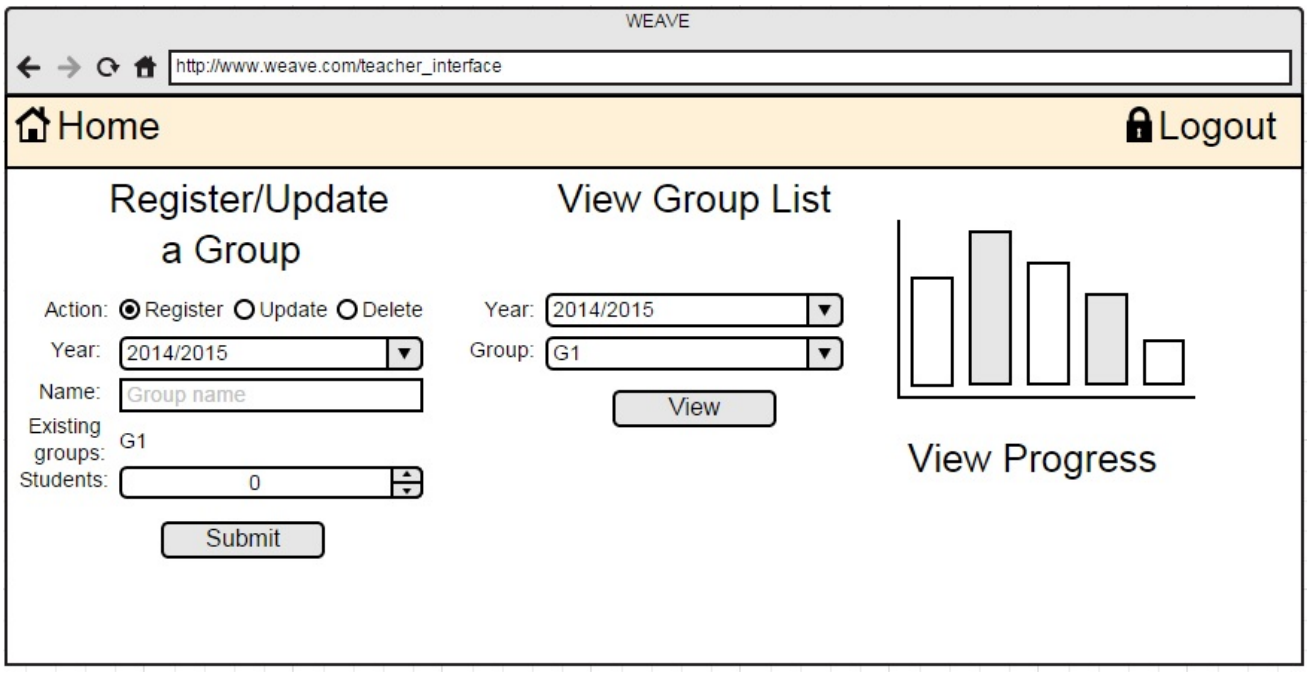


Figure 4.6: A wireframe for the home page of the teacher interface (when the teacher is logged in).

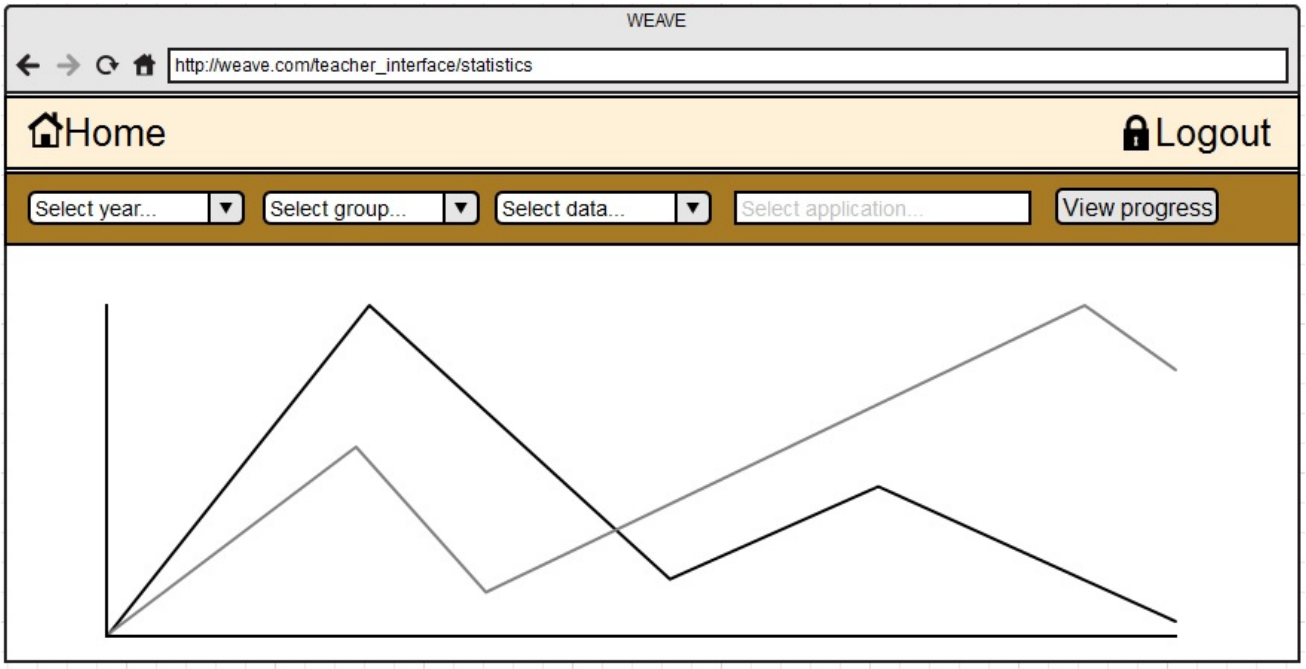


Figure 4.7: A wireframe for the page showing the progress of students.

4.8.2.3 View Group Page

Teachers are able to see the randomly generated pupil ids for a class in the `View Group` page. Because managing login details with school pupils is known to the project supervisor to be a problematic issue, this page was designed in close consultation with teachers to ensure the simplest way for them to distribute pupil ids. Furthermore, it is designed to ensure that the anonymity of data gathered by the system is complete.

The pupil ids are provided in a table with empty columns for the pupil name and two identical columns with

the pupil ids. There are two expected modes of use:

1. Teachers print this group sheet and fill in the names of their pupils by hand. This avoids any potential problems with storing identification information in the system. Teachers can then cut one of the columns for the pupil ids, further cutting the column into the individual pupil ids and hand them privately to each pupil. The teacher keeps a paper record of which pupil has which id.
2. The system is designed to enable teachers to copy and paste the pupil id column into a spreadsheet containing the pupil names. This can be printed as before, or projected for pupils to read off.

4.8.2.4 View Progress Page

Most of the design decisions for the different types of graphs are explained in Section 4.6 above. The wireframe for this page is shown on Figure 4.7. The teacher needs to select the particular group and the type of data they are interested in. If there is no data for that selection or the selection is invalid, an appropriate message appears on the screen. Otherwise, a graph is shown. This graph is downloadable to enable saving the data for progress at different points in time and could be used for comparison by the teacher.

Chapter 5

Implementation

This chapter outlines the most important and interesting aspects of the implementation of WEAVE.

5.1 Technology Choices

This section describes the considerations taken into account during the process of selecting the technologies for this Level 4 project. It then follows with the list of technologies chosen based on these considerations and the constraints of this project.

5.1.1 Considerations taken into account

Due to the size and the nature of this project, the technologies for it were selected carefully based on the following criteria:

- Scalability. WEAVE is intended to be used in around 400 schools in Scotland by thousands of pupils- and if successful, then internationally. This is why the system being scalable is one of the most important considerations taken into account.
- Ease of development. This criterion is important due to the constraints of the Level 4 project (mostly the time constraint). It was preferred that the selected technologies allow quick and easy implementation and testing.
- Maximum coverage of the requirements. The selected technologies should allow the satisfaction of the requirements in an efficient way. They should also allow flexibility for any possible changes.
- Abstraction of other components. Separating different components is a well-recognised software engineering practice and would improve the maintainability of the code.
- Sufficient documentation. Learning and improving skills in the chosen technologies is one important benefit of this project so the presence of clear and detailed documentation is desired.
- Prior experience. Having good knowledge in the technologies used will result in a smaller learning curve and maximum effort could be concentrated in development, testing and improvement of the product.

	PHP	Ruby on Rails	.NET	Django
Is it scalable?	yes	yes	yes	yes
Does is allow rapid application development (RAD)?	yes	yes	yes	yes
Can it cover all the requirements for this project?	yes	yes	yes	yes
Does it impose separation of concerns?	no	yes	yes	yes
Is sufficient documentation available?	yes	no	yes	yes
Do I have prior experience?	no	no	no	yes
Is it open-source?	yes	yes	no	yes

Table 5.1: A table showing the options and how they satisfy the criteria used for answering the question "What web application framework would be most suitable for this Level 4 project?"

5.1.2 Choices

5.1.2.1 The Backend

The choice for a web application development framework was of crucial importance for this project. To facilitate making the most suitable choice of framework, the Questions-Options-Criteria design model was used [8]. Table 5.1 shows the question (in the caption), the different options and the criteria considered for this project. Based on the maximum satisfaction of the criteria in the Table, the Django web framework , written in Python, was selected for implementing the backend component. Firstly Django will allow scalability due to the fact that it supports many database engines and switching between them only requires changing the settings of the project. This project will use the default database engine- SQLite because there will be no need for a more powerful one for the time being. However, as WEAVE evolves and becomes more widely deployed, it will be trivial to change to a more powerful engine. Secondly, Django allows rapid application development (RAD). It has support for many libraries which ensures that all the requirements for this project can be met. Django imposes separation of concerns which is discussed in more detail later in this chapter. Both Python and Django are very well-documented and developing web-based applications using them has turned into an enjoyable activity thanks to the award-nominated beginner's guide to web development with Python and Django How to Tango with Django[1] available online. Furthermore, I have sufficient prior experience coming from studying Python in the first year at the University of Glasgow and the Django framework in the Distributed Information Management 3 module. Finally, Django is open-source so there was no need to pay for the software used to develop this project.

5.1.2.2 The Web Interface

There are two groups of technologies for the web interface- for the server and the client side.

5.1.2.2.1 Server Side The Django web framework has sufficient support for serving clients' requests. This is achieved via direct communication with the database. Django allows clear separation between presentation and business logic due to the usage of the *Model-View-Template(MTV)* software design pattern which guarantees better maintainability and readability of the code. This pattern is described in greater detail in Section 5.2.1.

5.1.2.2.2 Client Side An important role in the selection of technologies for the client side of the web interface plays the compatibility with the browsers which are expected to be used in schools, mainly older versions of Internet Explorer, Google Chrome and Mozilla Firefox. Typically, pupils in schools are constrained to use the

[1] `http://www.tangowithdjango.com/`. Last accessed March 25th, 2015.

browsers that are installed on the school machines and acceptable appearance and behaviour for these browsers is crucial.

HTML5. This is the markup language chosen for the generation of the web pages. It is preferred to other markup languages due to the multiple benefits it has. Firstly, HTML5 provides an easy access to contents and elements which helps for design and debugging purposes. Secondly, it allows for writing of cleaner code where style and content are separated. Last but not least, HTML5 supports excellent cross-browser compatibility. It was preferred over XHTML[2], which was the other option under consideration, because HTML5 offers more flexibility.

CSS. Cascading Style Sheets (CSS) is used for defining the appearance of the web pages. It enforces separation of concerns between the HTML elements and their presentation. Furthermore, the use of CSS allows the control and flexibility over the appearance of different elements and results in a cleaner HTML code. An alternative considered while making this choice was Less[3]. It adds features to the CSS language and makes it more maintainable. However, it would have involved some initial learning overhead and due to the time constraints of this project it was decided to focus the predominance of the time on other parts of the implementation instead.

JavaScript. Being open-source and supported by the majority of browsers, this scripting language is used to deal with the interactions of the client and the webpage. Alternatives for this choice of technology were not considered due to the simplicity and powerfulness of this language for its intended purpose.

JQuery. Having the same benefits as JavaScript, JQuery is an excellent solution for simplifying client-side scripting. It is used for defining the behaviour of different components on user interaction with the website. It allows easy manipulation of HTML elements and removes cross-browser issues. Furthermore, it has a very large community and it is easy to understand, as opposed to the alternatives considered, which include Kendo UI[4] and Wijmo[5].

AJAX. Asynchronous JavaScript and XML (AJAX) is used for sending asynchronous requests to the server side avoiding the need for reloading the web page. This reduces the network overhead and the behaviour of the application feels closer to a desktop one. The alternative to using AJAX would have been to load pages synchronously. However, this would make the interaction cumbersome and slower.

Bootstrap[6]. This is the most popular framework for developing responsive design of web applications. This is a combination of HTML, CSS, and Javascript code that is designed to facilitate building of user interface components. WEAVE is intended for school computers and the screen sizes may vary across schools. Furthermore, it is not guaranteed that students will use full screen size at all times of interaction. Due to the nature of WEAVE, students may need to open a different window with more information needed to enable them to solve a problem. These needs lead to the responsiveness of the application being crucial. An alternative considered was PureCSS[7] which provides a set of small, responsive CSS modules that would load faster. However, Bootstrap was preferred because of the larger community and the greater number of features it offers.

Font Awesome[8]. This is an open-source library used to simplify the user interface through the use of familiar icons for visualising possible means of interactions with the webpage.An example icon used across all the pages is the Home icon which users of websites are familiar with as a means to navigate to the home page. There are many open-source libraries which could have been chosen for this purpose. These include Entypo[9] and

[2] `http://xhtml.com/`. Last accessed March 25th, 2015.
[3] `http://lesscss.org/`. Last accessed March 25th, 2015.
[4] `http://www.telerik.com/kendo-ui`. Last accessed March 25th, 2015.
[5] `http://wijmo.com/`. Last accessed March 25th, 2015.
[6] `http://getbootstrap.com/`. Last accessed March 25th, 2015.
[7] `http://purecss.io/`. Last accessed March 25th, 2015.
[8] `http://fortawesome.github.io/Font-Awesome/`. Last accessed March 25th, 2015.
[9] `http://www.entypo.com/`. Last accessed March 25th, 2015.

Typicons[10]. Font Awesome was chosen on the grounds of providing better looking icons.

HighCharts[11]. This free library is chosen because it allows the easy creation and control of interactive charts which are needed for the teacher interface of WEAVE. It has been chosen mainly because it provides many pre-built graphs, such as the line graphs and the column charts needed for viewing the progress of pupils. Another benefit is that the library is compatible with older browsers like Internet Explorer 6. This is needed because in schools across the UK older browsers are used predominantly. A library which would bring more powerful features would be D3[12]. However, it was not chosen because it does not work well with older browsers.

5.2 Connecting the Different Tiers

As described in Section 4.7, the architecture of WEAVE involves three tiers- the presentation tier, the Django middleware and the Data tier. The Django middleware is split into two distinct sub-tiers- one to serve the communication between the client and the database, and one to deal with the imports of worked examples created by the author interface of the old system. For the purpose of this chapter, this architecture model will be split into two parts. The first part will represent the components in the purple area shown on Figure 4.3. These components realise the communication between the client and the server. The second part will represent the components responsible for the transformation of the worked examples created by IWE into database objects. These components are coloured in green on the same Figure.

5.2.1 Client-Server Communication

The different components responsible for the Client-Server communication are glued together via the Django variation of the *Model-View-Controller (MVC)* design pattern- *Model-Template-View (MTV)*.

5.2.1.1 Model

The model represents the data stored in the system. There are two types of data objects- one for the worked examples and one for the usage data.

The objects storing the worked examples are strongly influenced by the structure of the XML files generated by the IWE author interface for holding these examples. Here, however, the relationships between objects are imposed by the use of foreign keys. For example, in the `Documents.xml` file, fragments are defined by their id and by the text of the fragment. `Processes.xml` file defines the steps for an example again by specifying both the id and the text of fragments. In this implementation, instead of the `Document` entity and the `Step` entity to be associated with a fragment via `CharFields` for the fragment id and the text of a fragment, fragments are defined as separate entities and documents and processes refer to them via foreign keys. This contributes to a much easier and reliable modification of worked examples due to the fact that if a change needs to be made to a fragment, this change will need to be made in one place only- the `Fragment` object.

The objects storing the worked examples and usage data for the worked examples are explained in detail in Section 4.2 and Section 5.3 respectively.

[10] `http://typicons.com/`. Last accessed March 25th, 2015.
[11] `http://www.highcharts.com/`. Last accessed March 25th, 2015.
[12] `http://d3js.org/`. Last accessed March 25th, 2015.

5.2.1.2 Template

Templates describe how the data is presented on screen. They are equivalent to the view in the well-known MVC design pattern. Each template is an HTML file defining the different elements to be rendered on screen. The style of these elements is defined via CSS. Depending on the user interaction with each of the interfaces, elements can be destroyed, hidden, created or modified accordingly using JQuery. Such calls are predominant in the teacher interface. For example, if the teacher wants to see the answers for a question in a particular example, a dropdown with the relevant questions appears upon selection of the worked example. If they wanted to see the total time a pupil worked on a particular example, the dropdown list of questions would be exchanged with a dropdown list of pupils belonging to the selected group.

The templates define alternative elements to be rendered depending on whether the user is logged in and if they are- based on their previous interactions with the system. There are different elements that can be rendered depending on previous interactions with the system. For example, the home page of the teacher interface shows register/log in sections when the teacher has not logged in, and if they have- areas to register/update/delete a group, view pupil list for a group and view statistics options. This is done via a template language condition:

```
{% if user.is_authenticated %}
    show the elements for areas to register/update/delete a group, view pupil list for a group,
    or view statistics options
{% else %}
    show text boxes to register/login
{% endif %}
```

Similar template language is used to iterate over the list of examples which are passed to the context when the main page of the student interface is rendered. More details on how the list of students is passed are provided in the description of the HTTP requests below.

There are elements occurring in all templates of the interfaces. For example, the navigation bar on the top of each page. To improve maintainability and to avoid too much repetition of the same code over and over again, the reoccurring content is extracted into a `base` template from which all other templates inherit.

5.2.1.3 View

Each view plays the role of the Python callback function for a particular URL. Views are the equivalent of the Controller in the MVC design pattern. Different parameters may be passed via the request made by the client. There are two types of requests:

- HTTP requests to render a page. An example of such request would be when the pupil clicks on a worked example to work on- the client will send a request for the page for this worked example. Each request has its context, containing information such as the clients machine, etc. Many pages require some information to be passed upon rendering. For example, when a pupil authenticates themselves, the list of the existing worked examples must be rendered on screen. The view responsible for serving this URL knows that this page requires this list, so it will send an ORM request to the database for all examples. It will add these examples to the context dictionary and respond to the client's request by rendering the template for the requested URL and passing the examples via the context dictionary.
- AJAX requests. These requests do not render another page but are used to get information from/sent information to the database. There are two types of AJAX requests served by the views:
  - GET requests. In these requests, the client asks for information from the database. Such requests are used mostly in the teacher interface when the teacher views the usage data of their pupils. The

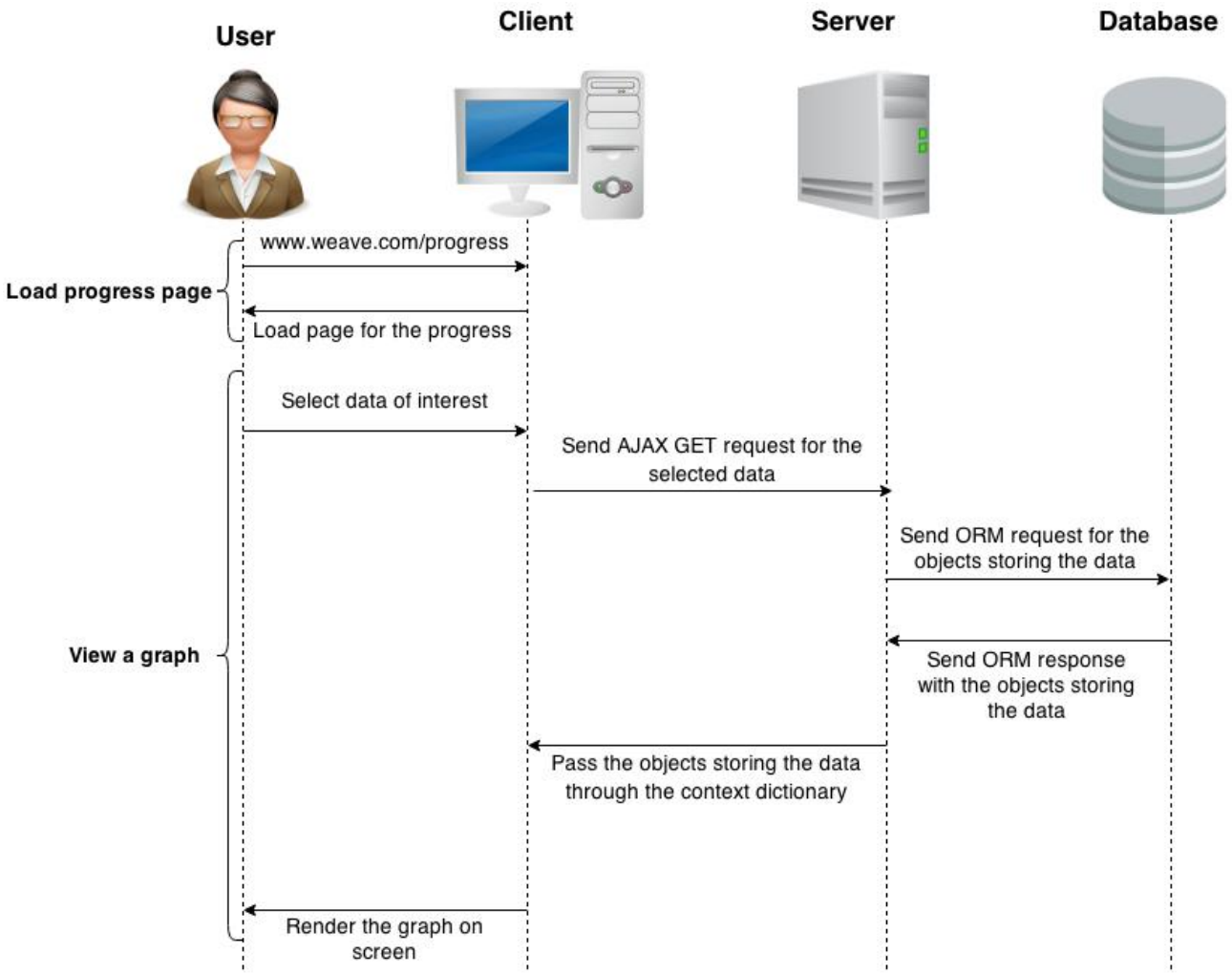


Figure 5.1: A flow diagram explaining the AJAX GET request when the teacher loads a graph with usage data.

requested information is passed in the form of a dictionary and the responsible view gets the required variables using the keys of this dictionary. A flow diagram showing this request is shown on Figure 5.1

– POST requests. These are requests to store some information in the database. Logging of usage data for the worked examples, such as the time spent on a particular step, is done via these requests. The response typically indicates whether the data was stored in the database successfully. A flow diagram showing this request is shown on Figure 5.2. More details how logging of data is implemented follows in Section 5.3.

5.2.2 Translation of the XML Elements into Database Objects

The translation of the XML elements into database objects is done via a population script which uses an XML element tree parser. Each element in this tree is represented by its tag and a dictionary of its attributes. There are two possibilities for the attributes of the element. They can either be attributes of the database object for the respective element, or references to other database objects which will be used to create a foreign key relationship between these objects. This parser uses the depth first search algorithm to traverse the elements in the XML files. The pseudocode for translating the `Processes.xml` file (in IWE, the steps for each worked example are

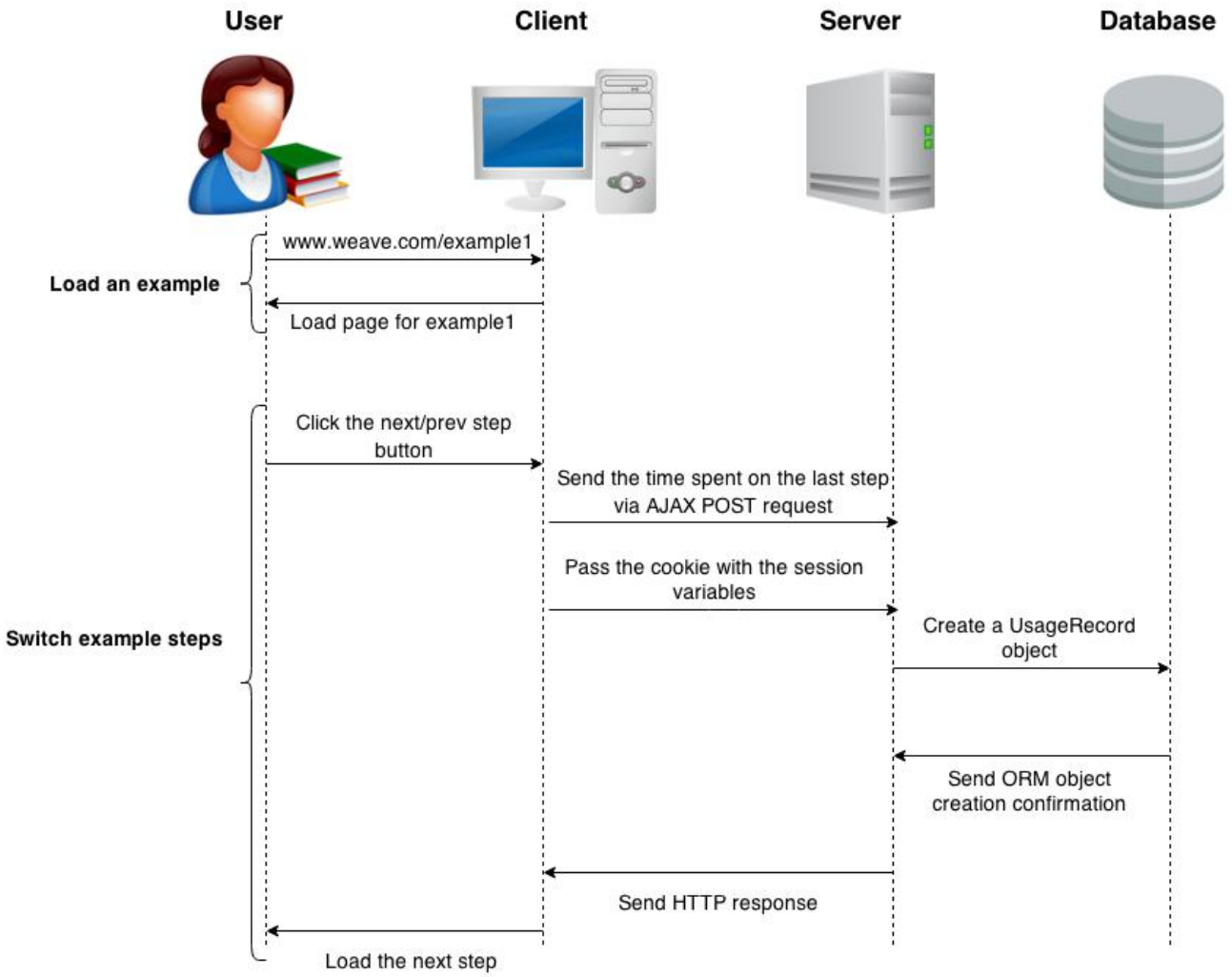


Figure 5.2: A flow diagram explaining the AJAX POST request when the pupil moves between steps of an example.

collected in a `process`) into the respective database objects is shown below. Please note that in this pseudocode `application` stands for worked example. This is the terminology of IWE.

```
open the XML file
initiate element tree parser on this file parser
get the root of the tree from parser processes

for process in processes
    get the name of the example this process is for from process attribute app_name
    get the application object with name app_name from the database model application

    for each step in process
        get the order of this step from step attributes
        create a step object with foreign key application and attribute order

        for each child in step
            if child has tag change
                get the fragment with the ID specified in the change attributes from the DB
                get the fragment with the document name in the change attributes from the DB
                get the operation to the fragment for this change from the change attributes
                create the change object with foreign keys step, fragment, document and operation
            else if child has tag explanation
                get the explanation text from the explanation attributes
                create explanation object with foreign key step and the explanation text as attr.
```

Please note that for simplicity reasons this preudocode describes the process for adding changes which involve showing/hiding/highlighting/unhighlighting of fragments and the presence of an explanation. It does not include the creation of changes which involve questions. However, these changes, as well as the rest of the XML files, are translated following the same logic.

From this pseudocode, it becomes clear that only relevant elements are stored in the database. For example, no objects were created for a `process`. `Processes` become redundant in this database organisation and they can be unambiguously represented via the `Step` objects instead.

Creating and updating objects are done via the Django `get_or_create` method and passing the primary key for this object. This call retrieves either a newly created object or an already existing object with the specified value for the primary key. When this object is retrieved from the database, its attributes are set to those coming from the attributes dictionary of the XML element. This ensures the correct creation or modification of the object and is the reason for the destructive update model of WEAVE.

5.3 Logging of Usage Data

Usage data is logged on every step transition. The data for each attempt of an example step is represented by a database object. There are two types of such objects to accommodate the two different types of steps (steps which involve showing/hiding/highlighting/unhighlighting of fragments of a document and showing an explanation, and steps which involve a question). The respective objects are UsageRecord and QuestionRecord.

On change of the step, an AJAX post request is done invoking the relevant method in the views depending on the type of the step. This POST request passes a dictionary with the necessary information for the data record:

- the example name, the step number, the time spent on this step and the direction for the transition for the step. This is the data sent for steps which involve changes of fragments and an explanation and is stored in a UsageRecord object; or
- the example name, the step number, the time spent on this step, the answer for the question and an indication whether this was a multiple choice question. This is the information sent for steps which involve a question in a QuestionRecord object.

To handle this request, the relevant view extracts all the usage data using the relevant keys of the dictionary passed by the AJAX call. However, this dictionary does not pass any information about the user this data is coming from. This information is accessed via session variables instead. When the pupils specify their details, these are stored as session variables. For example, to store the pupil ID as a session variable when the pupil specifies their details, the view serving the AJAX call will include the line:

```
request.session['pupil'] = pupil_id
```

When the view for handling logging of data is invoked, it checks the session variables for the teacher id, group id and pupil id .This is done via a simple call such as:

```
pupil_id=request.session.get(pupil, None)
```

None is the default value to return if no such session variable exists. After checking the session variables for the teacher id, group id and pupil id, the method adds the ones that were present to the data record and saves the record in the database.

5.4 Resizing of the Worked Examples Panels and the Explanation Area

A major implementation challenge for this Level 4 project was to satisfy the requirement to make the panels for the text of the worked examples, as well as the explanation panel, resizable. Difficulties arose due to two requirements for the system:

- The system must support worked examples which contain any number of panels for worked example text (not only one or two).
- Both horizontal and vertical resizing were required.

The JQuery `resizable` plugin provides the option to specify that an element is resizable. When specifying this, it can also be specified that when the resizable element is resized, other selected elements are resized as well. This can be done by using the `alsoResize` option supported for the `resizable` elements. The code for making an element with id *panel1* resizable and to specify that an element with id *panel2* needs to be resized as well on resizing of *panel1* would be as follows:

```
<script>
    $(function() {
        $(`#panel1`).resizable({alsoResize: `#panel2`});
    });
</script>
```

However, this would mean that when making one panel wider, the other panel also becomes wider. By small addition to the plugin code, a similar option to resize the second element in the opposite direction could be added- `alsoResizeReverse`. To implement this option, it would have been required to change the sign in calculations done by the respective function. The above code would then look like this:

```
<script>
    $(function() {
        $(`#panel`).resizable({alsoResizeReverse: `#panel2`});
    });
</script>
```

The following subsections will describe the problems using this approach.

5.4.1 Vertical Resizing of Worked Examples Panels

The solution above could work for resizing the panels vertically if there were two panels for the examples only. Using it for more than two panels would behave in an undesired and unexpected way due to too much `alsoResizeReverse` dependency between the panels.

Another JQuery plugin which was found to behave better in this situation is the `colResizable` plugin. It treats the area of the worked examples as an HTML table where the panels are different columns which can be resized. This plugin was used for the vertical resizing.

5.4.2 Horizontal Resizing of Worked Examples Panel

The `resizable` JQuery plugin was ideal for the horizontal resizing of the panels. However, it was incompatible with the `colResizable` plugin used for the vertical resizing of the panels. This is why, the height of the panels and the explanation are recalculated and set manually on each resize event. This is the code:

```
<script>
    $(`#explanation_container`).bind(`mouseup`, function(e) {
        var interface_height = $(`#interface`).height();
        var container_height = $(`#explanation_container`).height();
        $(`#panel_container`).height(interface_height - container_height);
    });
</script>
```

What this code does is when the user is finished with the resizing of the explanation panel, it sets the height of the worked examples panels to be equal to the difference between the total height dedicated to the worked examples panels and their explanation (referred to as `interface` in this code) and the height of the explanation panel.

5.5 Showing the Text of the Worked Examples in the Panels

As described in Chapter 2 and Chapter 4, the documents containing the text of a worked example are split into many fragments. These documents are shown in separate panels on the viewer. These panels are in the form of HTML `div` elements. It is often the case that more than one fragment need to appear on one line in these `div` elements. This is why the fragments needed to be represented by inline elements rather than block elements in HTML. Inline elements allow multiple elements to be placed next to each other, while block elements are placed below each other. `span` is the inline element chosen for representing each fragment. Please note that any inline element could have been chosen without observing any differences in the behaviour.

Due to the constraint that some fragments need to be inline with other fragments, a slightly undesired behaviour of the fragments arose when they do not fit in the width of the panel for a particular document. Words of these fragments can be split between two lines in unexpected parts. However, this is a limitation due to the use of HTML elements and their behaviour.

5.6 Dealing with Concurrent Viewing and Modification of Worked Examples

Due to the web-based nature of WEAVE it is possible that pupils might be working on examples at the same time a new version of these examples is being uploaded. To avoid inconsistency in such situations, when the request for the worked example page is served, all the steps for that example are retrieved and sent back to the client in one response message. This means that the response would be heavier (it will transfer more information at one go) than retrieving the fragments of a worked example when required for a particular step, but it will ensure that the pupil will always see one version of the worked example only. This is how the atomicity is ensured across the system- the pupils see the entire new version or the entire old version of the example.

Chapter 6

Evaluation

This chapter describes the approaches for and the key results obtained from the evaluation of WEAVE.

6.1 Goals for the Evaluation

The main questions to be answered during the evaluation are:

- **Is the system usable by pupils?** The evaluation will examine how good the student interface is in communicating the worked examples effectively. It will make conclusions of whether the student interface facilitates or hinders viewing of worked examples and it will indicate potential poor design decisions for this interface to be improved in the next version of the system.
- **Is the system usable by teachers?** The evaluation will examine the extent to which the design decisions for the teacher interface support teachers in their goal to monitor the progress of their pupils and it will indicate potential poor decisions for this interface to be improved in the next version of the system.
- **Is the usage data of pupils presented in a clear and understandable way for teachers to help them to better understand the difficulties of their pupils?** The evaluation will examine how well the progress information for the pupils is visualised on the graphs for the teacher interface and will indicate any poor design decisions for the visualisation of data to be improved in the next version of the system.
- **Is WEAVE easily deployable in schools?** The major motivation for this project is to create an easily deployable tool to be used in schools. Even though no installation will be needed for the application, it is possible that schools block the website due to various access policies. This evaluation will examine how easy it is to start using WEAVE in schools.

6.2 Methods of Evaluation

Four different types of evaluation have been decided upon in order to answer the questions above as thoroughly as possible.

- Usability Evaluation conducted with 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.

- 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 achieved its goal to be easily deployable in schools.
- Heuristics Evaluation conducted with 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"[1].
- Practical Evaluation with the secondary school teachers Mr. Peter Donaldson and Mr. Craig Henderson, who will use WEAVE as a teaching technique with their pupils to evaluate the system more thoroughly in achieving its goals, in other words- is it easy to use WEAVE in schools, is it well-accepted by pupils and is it helpful for them as teachers to better understand the problems of their pupils.

6.3 Usability Evaluation

6.3.1 Purpose of the Evaluation

The focus of this evaluation is to assess the quality of the user's experience at the time of interaction with the system. To ensure that the evaluation results are not affected by the contents of the worked examples but they rather provide useful insight for the design of the user interfaces, it is conducted with both participants with and without experience in Computing Science. Comparison of the results of the two groups of participants will result in higher confidence in the conclusions drawn from this type of evaluation. Both types of Usability Evaluation- with inexperienced and experienced participants- were conducted with five people. The choice of number of people is made on the grounds of literature which suggests that five participants are enough to discover around 80% of the usability issues in a system [19].

6.3.2 Usability Evaluation with Participants without Experience in Computing

6.3.2.1 Structure of the Evaluation

The participants were encouraged to use the tool for about half an hour and then answer two sets of questions for the system. During the evaluation, these participants were advised to not try and understand the meaning of the worked examples they were working on and should not worry if they were not familiar with the terminology involved in these examples.

The evaluation was conducted in room 208 in SAWB in the University of Glasgow, at times when no other people were present in this room. Participants were briefed by the evaluator with information about the context the application is intended for, as well as the main functions of the system. They were then given an information sheet with instructions about the goals they need to achieve using the system and one of the sets of questions they need to answer after the evaluation. This information sheet is shown in Appendix X. The reason for providing the first set of questions in advance is so that participants are aware of them and they can take notes during the evaluation if they desired so.

In short, participants needed to achieve 3 goals:

1. Pretend they are a teacher and register a group for their pupils.

[1] `http://www.tangowithdjango.com`. Last accessed March 25th, 2015.

2. Pretend they are a pupil of the teacher they registered for the first task and work on an example of their choice.

3. Pretend they are a teacher again, and view the progress of the pupil from task 2.

During their interaction with the system, participants were encouraged to think aloud. The evaluator was quietly observing them, taking notes on how they were interacting with the system, as well as on their"thoughts". The evaluator paid particular attention to signs from the user interaction providing insight on any of the following categories:

- Intuitive design. Do the users understand the architecture and the navigation of the application?
- Ease of learning. How fast did participants who never used WEAVE accomplish their goals?
- Errors and recoverability. What errors did the participants make and how quickly did they recover from them?
- Satisfaction with the system. How satisfied with the user interface are the participants?

After the participants completed the goals for the evaluation, they answered the questions in the information sheet given to them at the start of the evaluation. They also rated the system using the well-recognised System Usability Scale (SUS). Literature suggests that SUS is a quick and reliable usability assessment method [2]. With this rating system, a score above 68 is above the average score for around 10,000 interfaces, generally viewed as an indicator of acceptable usability.

In general participants did not encounter any significant difficulties while using the system. A few of them were uncertain what details they were supposed to enter for task 2. However, the reason for this confusion is that it is not easy to play a role and imagine different situations- first being a teacher, then being a pupil, then a teacher again- and it is natural to be confused. On these occasions, participants were given minimal guidance, due to the fact that when real teachers and pupils will be interacting with the system, they will be in a real world scenario, where such details will be absolutely clear.

6.3.2.2 Results

6.3.2.2.1 Participants Thoughts and Actions Having read the instructions at the start, participants were clear what their goals were.

For the first task, participants needed to register, and later update, a group. They found the structure of the main page of the teacher interface very logically arranged. Their attention was immediately drawn to the large area for creating a group. Most participants commented that they really like the message, which appears on screen to tell them if the registration of their group was successful. What confused some participants was whether the number of pupils selected on update of a group will be added to the number of existing pupils for the selected group, or they needed to enter a new number for the total number of pupils for this group. To make sure they enter the correct number, they navigated to the question mark providing help which is located next to the input field for entering the number of pupils. The majority of the participants tried multiple times to click on this question mark and were expecting a pop-up with information to appear on screen. It was difficult for them to notice that the help text appears on hovering with the mouse cursor over, but without clicking on the area.

For the second task, participants needed to enter pupil identifying information and work through an example. The participants who read the tutorial found it useful and very easy to read. In general, they did not have

any problems with the examples they chose to work on. One of the participants felt unsure of whether they selected the correct answer to a multiple choice questions asked by the system. There were comments from some participants that they would want to see some headings for the different panels showing the examples. One of the participants did not answer one of the questions and they were unsure how this was recorded in the system. The automatic scrolling to newly appeared content of the worked examples brought different opinions. Here are quotes from two of the participants:

"I don't like it how when I move from different steps, the scrollbar moves to show me the new parts of the example. I wish it stayed on the last point in the text I reached."

"I like the animation showing the new text as it appears. It definitely draws my attention. I don't think I would have scrolled to the bottom myself so I could have missed something important there."

For the third task, participants looked at usage data they produced during their interaction with the student interface. The names of these graphs did not lead to their immediate understanding what the graph would show and participants wanted further explanation about each type of data to be provided. One of the participants was really into the role of a teacher and here is what they said when looking at the graph for the average time students spent at steps of an example:

"I wish I could see the total time students spent at this example so that I know whether examples of this sort would be suitable for in-class activities or for homework and I can make sure pupils are not under-/over-loaded with work."

Shortly after this comment, the participant opened another type of progress information and made a new comment:

"Oh, this is great! This is the information I wanted to see earlier. It makes more sense to be here!"

Most participants were unsure what the question mark and the number shown in the brackets for each step in the Average/Student time graph meant.

Even though some of the features of the graphs were not apparent, participants indicated that after a more detailed exploration of these graphs, they are very confident with the way they are presenting the information. They like it how a lot of information is visualised without the graphs being difficult and overloaded with everything on screen. In addition, participants were impressed that the system allows them to print these graphs.

6.3.2.2.2 SUS Score The average SUS score for the pupil interface is 90.5 and for the teacher interface is 83. This score is based on the individual scores given by each participant for each interfaces. The scores the participants gave on each question in the SUS questionnaire are attached in Appendix X.

6.3.2.3 Discussion on the Results

The confusion about the number of pupils a group will consist of on update of the group comes mainly from the fact that the participants in this evaluation are not teachers in schools and do not fully understand the context the system is going to be used in. The true users of the system will only want to update a group if more pupils have arrived to their class. They would be aware that only the new pupils would need a pupil id generated by the system so they will know that on update of the group, they need to enter only the number of pupils that are new to the group they are updating.

Comments which are made for the page for viewing examples are split into two groups-comments related to the user interface and how the looks and feels of the system affects the user interaction, and comments related to

the content of the worked examples and their presentation.

The issues with the user interface of WEAVE are minor. They would require adding more information about the current status of the system. Due to the small number of participants in this usability evaluation, it is still unclear whether scrolling automatically to the new contents of a worked example is a good feature of the interface. Further evaluation will be needed to clarify this.

Features related to the worked examples and their presentation are to a great extent closely restricted to what IWE currently provides. For example, being able to add meaningful headings for the different panels of an example cannot be done unless these headings can be added in the author interface on creation of the example. The student interface of IWE uses the name of each document as a heading for the panel containing it, but these names are often not very informative and could result in confusion rather than being helpful to the pupil. Being able to distinguish the correct answer to a question is also highly desirable for this kind of an application both for the student and for the teacher interface. However, this answer needs to be marked as correct in the author interface of IWE so WEAVE has no means of dealing with this at the moment. These comments will be addressed in a next version of the tool when author interface will be added to WEAVE.

The analysis of the results for the progress page in the teacher interface indicates that the way this data is presented to teachers would require some initial learning overhead on how to interpret the data. However, this overhead is not too large and after this initial learning stage the data will very easy to analyse.

6.3.3 Usability Evaluation with School Teachers in Computing Science

6.3.3.1 Structure of the Evaluation

This evaluation was rather informal due to the fact that the time the evaluation was conducted in was a very busy period for teachers and no suitable time could be arranged to do the evaluation face-to-face. Participants were sent an email with instructions written by the project supervisor. This email is attached in Appendix X. Participants were not sent the detailed information sheet used for the usability evaluation conducted with the inexperienced participants because it was likely that teachers would have been discouraged to read it due to its length and because they did not need to play roles in imagining that they were teachers. During this evaluation, participants were interacting with the system on their own. Some chose to work with the student interface only, while others tried both the student and the teacher interface. After the interaction, teachers completed a questionnaire with questions for the usability of the interfaces and the ease of deployment of the system in schools, as well as the SUS questionnaire for both interfaces.

6.3.3.2 Results

6.3.3.2.1 Participants' Answers to Questions In general, teachers are very enthusiastic about the idea behind WEAVE and they would use it as a teaching technique with their classes if suitable examples were provided. One teacher comments that they would have benefited from better knowledge of what they were expected to do at each task. Requests for functionality, which appeared in the results from the usability evaluation with inexperienced participants as well, are the need for headings for each of the panels containing the worked examples and for indication whether the correct answer to a question was selected. A really interesting suggestion by one of the participants is to provide a feedback option to allow users to make any suggestions for the improvement of the interface as they interact with it. This would provide constant and much wider feedback for the usability of the system. Another interesting suggestion from an educational point of view is to set a minimum time to spend on a question to encourage pupils to think about the answer of this question more thoroughly, rather than rushing to the next step. Also, some communication between a teacher and their pupils through the system would be

a good feature because pupils would be able to ask questions about the examples more privately. There are no logistical problems found for the system (no website blocking, no machine limitations). According to teachers, the system can be very easily deployed in schools.

6.3.3.2.2 SUS Score The SUS score given by this set of participants for the student interface is 73.8 and for the teacher interface is 72.5.

6.3.3.3 Discussion of the Results

Due to the evaluator not being present on the interaction of the participants with the system and to the informality of the evaluation itself, it is not clear what might have affected the results. A possible reason of why the SUS scores are significantly lower than the scores given by the other participants of the usability evaluation is because the teachers did not receive as much background information for the system, and it was not entirely clear to them what they are evaluating. From their answers, whose full text can be found in Appendix X, it can be seen that most teachers are commenting on the value of the worked examples provided and any issues with them rather than providing a more general insight on the actual appearance of the interfaces. Furthermore, the SUS score for the teacher interface was calculated with data from two teachers only, due to the fact that other teachers did not work with the interface. One of the teachers gave a score much greater than the average of 68- 87.5, while the other teacher's score was significantly lower - 57.5. The reason for this scoring needs to be investigated further and no conclusions can be drawn with the current data from this group of participants until a more controlled and better-organised evaluation is conducted.

6.4 Heuristics Evaluation

6.4.1 Purpose of the Evaluation

The Heuristics Evaluation of this project was conducted with two experts in Django application development from the Computing Science department in the University of Glasgow- Dr. Leif Azzopardi and Mr. David Maxwell. The main purpose of this evaluation was to have the interface compared against well-established usability principles and to identify any potential issues with the usability of the system. These principles are listed below, together with a short explanation for each.

- Visibility of the system status. This principle evaluates whether sufficient feedback for the systems status has been provided on interaction with the system.
- Match between system and real world. This principle assesses the suitability of the terminology and the icons used in the system.
- User control and freedom. This principle evaluates the extent to which the user feels restricted by the system to achieve their goals or to navigate through the website.
- Consistency and standards. This principle assesses how consistent the terminology and the look and feel of the system are.
- Error prevention. This principle evaluates the techniques used to reduce the places in the system where the user could enter an invalid input.

- Flexibility and efficiency of use. This principle provides an insight on how self-explanatory the interface is and whether any features for improving the efficient use of the system have been provided.
- Aesthetic and minimalist design. This principle evaluates the simplicity of the design while still being able to provide all the functionality required.
- Help users recover from errors. This principle assesses how informative the error messages provided in the system are.
- Help. This principle evaluates the quality and sufficiency of the help provided in the system.
- Navigation. This principle assesses the means of navigation across pages on the website.
- Structure of information. This principle assessed how well and logically structured different parts of the interfaces are.
- Physical constraints. This principle assesses the amount of effort required during the interaction with the system.

6.4.2 Structure of the Evaluation

The participants in this evaluation were given the same information sheet and instructions used for the Usability Evaluation with the inexperienced in Computing Science participants. They interacted with the system for about half an hour. In the end they were given an additional set of questions related to the heuristics listed above. This set of questions can be found in Appendix X. Mr. Maxwell's written answers are attached in Appendix X. Unfortunately, due to limited time dedicated for this evaluation, Dr. Leif Azzopardi was not able to answer these questions in writing. However, he expressed his general opinion about different aspects of the user interface and suggested some areas for possible improvements. His insights are taken into account in the discussion of the results below.

6.4.3 Results

This section summarises the results from the heuristics evaluation of both interfaces.

- Visibility of the system status. The evaluators concluded that the feedback given to the users during interaction with the system is informative and understandable.
- Match between system and real world. The results from the heuristics evaluation show that there is a good match between the system and real world. The terminology used is not an issue for users, the interface presents the information in a simple, natural and logical order and the icons used across the pages (the arrows for controlling the steps, the home icon for navigating to the home page, etc.) are appropriate for their purpose and easy to understand.
- User control and freedom. According to the evaluators, interactions with the system were clear and were not restricting them to achieve their goals.
- Consistency and standards. Generally, the controls for the system are consistent. However, the results from the evaluation indicate some stylistic issues that, if fixed, could improve the experience of the user. These include following a consistent colour scheme and size of different elements in the web page.
- Error prevention. It has been concluded that the system input fields are suitable for their purpose and they are chosen well to reduce the error rate as much as possible.

- Flexibility and efficiency of use. For this heuristic the two evaluators had significant disagreement in their responses. One way the student interface can be used more efficiently is by taking advantage of the shortcuts for controlling the steps of an example. Transitions between steps can be done either by clicking on the arrows provided on the screen or by using the arrow keys on the keyboard. This is explained in the short tutorial on the main page of the student interface. One of the evaluators did the tutorial before doing an example. He, therefore, was aware of the shortcuts provided in the system and rated positively this heuristic. The other evaluator pointed out that "nobody reads tutorials" and was unaware of the shortcuts the system provides so he rated this heuristic low. In addition, the same evaluator concluded that the system does not present the functionality in the most efficient way possible. For example, when viewing progress for a class, the teacher needs to make many interactions with the system until they get any progress information.

- Aesthetic and minimalist design. This heuristic is another disagreement point between the two evaluators. One of the evaluators found the design very simple to use and the different aspects of the system obvious and clearly labelled. However, the other evaluator suggested that the information (especially the one on the main page of the student interface) could have been presented in a better-looking and simplified way. For example, a greeting message could appear on screen when the pupil enters their details, as opposed to plain message confirming their identity.

- Help users recover from errors. It has been concluded that the messages appearing on screen on invalid interactions with the system are very informative and helpful to users.

- Help and Documentation. The results from the heuristics evaluation show that the help provided on hovering the question marks at places across the system is not obvious. Furthermore, as mentioned earlier, one of the evaluators commented that very few users read tutorials unless they are encouraged to do so. Both the help and the tutorial in the system will need to be more visible to the users for the next version of the system.

- Navigation. The navigation through different parts of the system was found simple and logical.

- Structure of information. Generally, the evaluation shows that relevant parts are placed together and in a logical manner. However, one of the evaluators indicated that too much information is shown on the main page for teachers. This assessment of this heuristic is closely related to the point he makes about the efficiency of presenting the information on screen. Combining both heuristics, a completely new structuring of the teacher interface needs to be done. General ideas about how the new version of the teacher interface would look after taking into account the feedback from the evaluators is shown in Appendix X.

- Physical Constraints. Cross-browser compatibility, which is a known issue for the system, affected to a great extent the answers of one of the evaluators. The page showing the worked example this evaluator chose to work on did not fit well on screen so he needed to scroll up and down the page to be able to switch between steps from the top of the page, and to be able to read the explanation for each step which is at the bottom of the page. When the system is used with a browser that is compatible with it (which was the case with the second evaluator) it does not take too much effort to work with the system.

6.4.4 Discussion on the Results

As pointed out in the summary of the results from the Heuristic Evaluation, there were disagreements for some of the conclusions made for certain heuristics. A highly possible reason for these points of disagreement is that one of the evaluators has a lot more experience and is rather a perfectionist. This is good because it is in agreement with the project developer's attitude and his feedback is taken very seriously. It is very unfortunate that this evaluation was not conducted at an earlier stage of the project so that these issues could have been addressed. However, both the project supervisor and the secondary school teacher this system was designed with- Mr. Peter

Donaldson- agreed that the correct functionality of the application has a higher priority for this project than its appearance. WEAVE was developed iteratively together with Mr. Donaldson, and he was generally happy with the appearance of both interfaces. The purpose of conducting the heuristic evaluation was to find ways in which this agreed interface can be improved for the future when the appearance of the system will be of greater priority. For the time being, it is satisfactory that the system is fully functioning and usable by both teachers and pupils. This by no means says that the comments about any poor decisions for the user interface will be discarded, but it rather means that they will certainly be addressed in the next stage of development of the application.

6.5 Practical Evaluation with Teachers in Schools

6.5.0.1 Purpose of the Evaluation

The purpose of this evaluation is to assess whether WEAVE is usable in the context it is intended for, as well as what is the current status of the application compared to what would be desirable for an application of this nature. In general, this evaluation will prove whether the concept and the implementation of WEAVE are suitable for use in schools. The current project is just the beginning of a much larger system. This evaluation aims to identify any areas for improvement and upgrades for the system so teachers participating in it were encouraged to be very critical. Although such criticality might be harsh for the developer of the system, it would be desirable any problematic areas to be identified and addressed on time to increase the potential of the application for the future because WEAVE will hopefully continue to grow after the end of this project.

6.5.1 Structure of the Evaluation

The teachers participating in this evaluation have agreed to use the system as part of their in-class activities with their pupils. They created a group for their class and they assigned their pupils with the ids generated by the system. Pupils then worked on an example specified by their teacher. In the end, pupils were encouraged to express their opinions about the student interface. Teachers selected some of their answers and summarised them for the evaluator. In addition, teachers answered the same set of questions used for the usability evaluation with the other teachers. However, their answers were expected to be more detailed, since they interacted with the system in a real-world scenario. In addition, these teachers were encouraged to think critically in terms of what the system is providing and what they would expect the system to provide if it was a real teaching technique in their school.

6.5.2 Results

Participants in this evaluation did not find the teacher interface intuitive enough. They required more instructions where to locate what they need. The page for viewing progress of pupils could be structured better to improve the user experience. Requests for the ability of the system to export the raw data shown on the graphs are made. Teachers would benefit from more flexibility for analysing the data in Excel using these csv files. Furthermore, the system could benefit from more control over the number of pupils for a class by enabling teachers to remove pupils from a group list. One of the teachers suggested that the group list with the pupils in a class could store the names of the pupils next to their ids and their privacy could be secured by encryption of those names with the password of the teacher. Suggestions of how to improve the user experience when interacting with the graphs, which show the usage data, were made. These can be simplified and can store the user choice rather than resetting the selections made every time the user changes the value for a group, for example. The visualisation of this data could be expressed in a clearer manner.

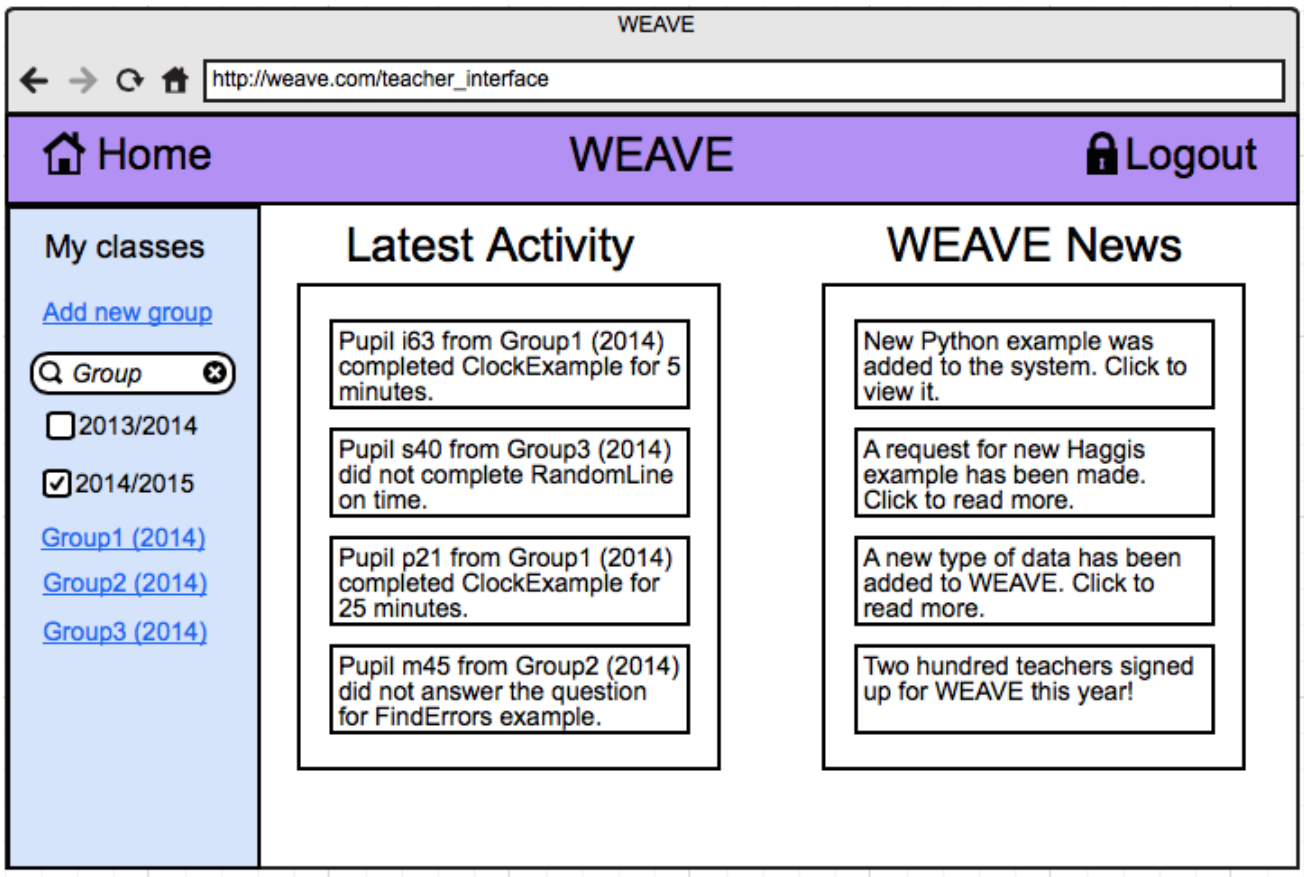


Figure 6.1: A wireframe for the new appearance of the main page for the teacher interface in WEAVE.

The suggestions for the student interface were mainly stylistic or regarding cross-browser compatibility.

Here are the concluding remarks of one of the participants:

"The potential of the system is really exciting and I hope that further work is done to develop it. Some of the features are really neat however several rounds of paper prototyping with different teachers need to be conducted in order to create a better experience in the teacher interface section. Individually there is nothing wrong with some of the data views but in practice making use of this data to improve learning and teaching would be difficult. This is something that many commercial systems dont necessarily get right either in version 1."

Despite the criticism, the evaluators like the idea behind WEAVE and would use the system if it is improved in the future. They concluded that it will be very easy to deploy WEAVE in schools and it was not blocked by Internet policies.

6.5.3 Discussion of the Results

As it was expected, the participants in this evaluation were rather harsh and this was highly appreciated by the developer of WEAVE. Soon after the end of this project, a new stage for the system is expected to begin. The criticism received for the first version of WEAVE will form the first set of requirements for this next stage.

Overall, what was expected to be the end product of this project has been satisfied and beyond. The criticism is mainly towards the teacher interface of WEAVE. It is important to note that the idea for this interface was developed halfway through the project as discussed in Chapter 2 and was not included in the initial set of

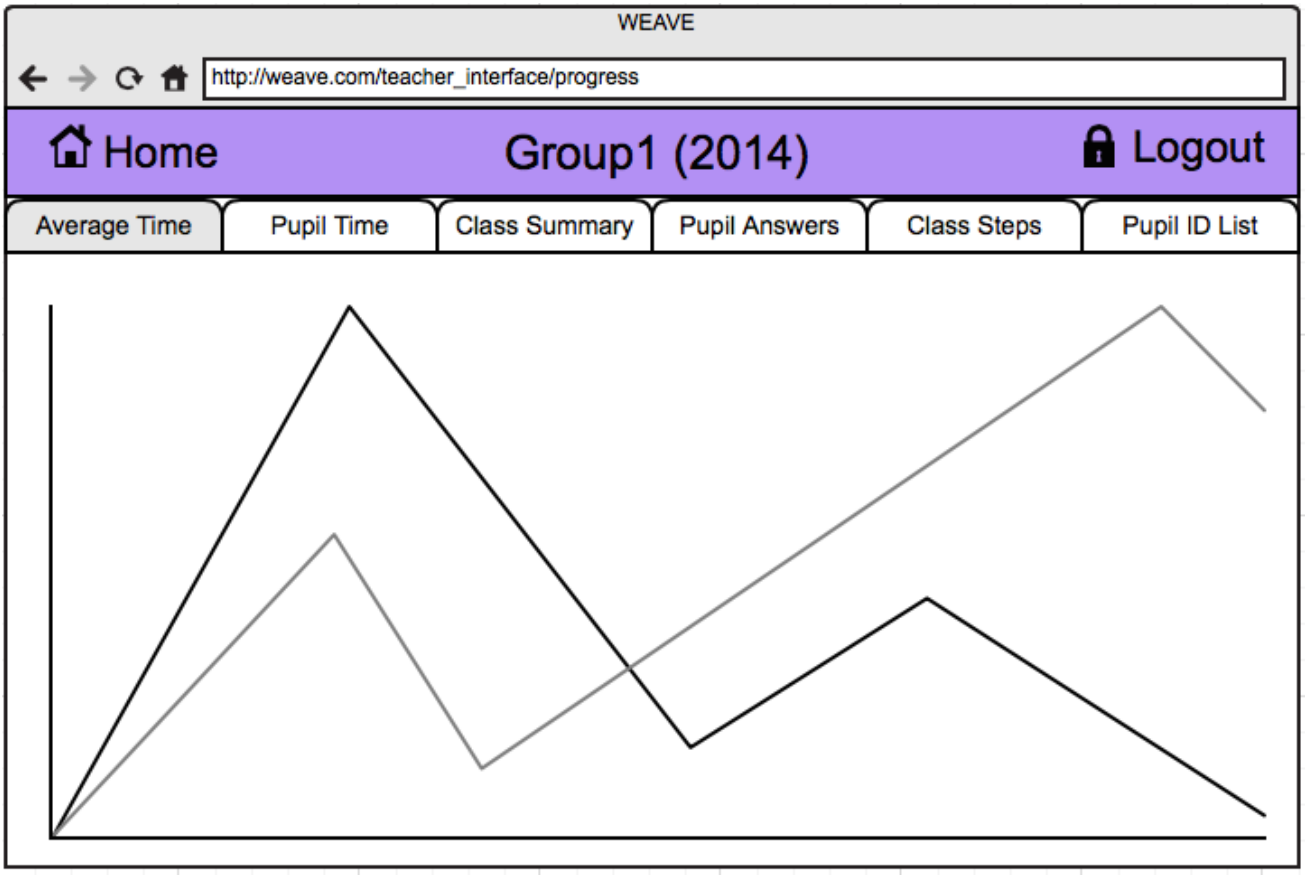


Figure 6.2: A wireframe for the new appearance of the page for viewing progress of a group in WEAVE.

requirements until then. Having to complete the teacher side of the system in a much shorter period of time inevitably resulted in having to make certain compromises. For this project it is a great achievement that it was used in schools and it is very exciting to get feedback from its intended users. The final feedback from Mr. Peter Donaldson says much about the achievements for this project:

"The system is something I would expect a team of people to have produced over a much longer period so the fact you've managed to get so much functionality as a student is really impressive."

The whole email from Mr. Donaldson is attached in Appendix X.

6.6 Conclusion

The conclusions drawn from each set of evaluation give answers to the questions listed at the beginning of this chapter and these are consistent across the individual types of evaluation.

- Is the system usable by pupils? Yes, the system has proven to be usable by pupils. There are stylistic details that could make the interaction with the system more satisfyable but apart from this no major issues have been identified.
- Is the system usable by teachers? Having a teacher interface could help teachers better understand their pupils. The interface provided in the current system is usable but it also needs to be improved to bring the expected value to the teacher.

- Is the usage data of pupils presented in a clear and understandable way for teachers to help them to better understand the difficulties of their pupils? To a certain extent yes, but the way this data is presented needs to be improved and the best way to do so is through iterative design in cooperation with teachers.
- Is WEAVE easily deployable in schools? There are no found difficulties found which would result in the inability to use WEAVE in schools.

Based on the evaluation, plans how to make the teacher interface more intuitive and efficient in presenting the information were immediately developed. General ideas about how the new version of the teacher interface would look after taking into account the feedback from the evaluators is shown on Figure 6.1 and Figure 6.2. Unfortunately, due to the late stage of the evaluation of this project, these have not been implemented in the system.

Chapter 7

Discussion and Conclusion

This chapter summarises the work that has been undertaken as part of this project and describes ideas for future work for the developed prototype.

7.1 Summary of the Project

7.1.1 Project Stages

Below are the different stages for this Level 4 project.

- A problem with installing previously developed software in school classrooms that could improve the techniques for teaching school-level Computing Science has been identified and a solution to this problem has been proposed.
- Through discussion with school teachers ideas of how to make the proposed software valuable to them as well were developed.
- The design of the web-based application to be used in schools by both teachers and pupils was formulated in close cooperation with a secondary school teacher.
- The implementation of this web-based system was undertaken.
- The evaluation of this system clearly indicates that it will be valuable for use as a new teaching technique in schools and teachers are enthusiastic about the future development and integration of this system.

7.1.2 Satisfaction of Requirements

This section discusses the extent to which the requirements for WEAVE are satisfied.

7.1.2.1 Functional Requirements

The system satisfies all of its functional requirements.

Firstly, WEAVE is able to easily upload worked examples created with the author interface of IWE.

Secondly, the teacher interface provides a means for teachers to group their pupils, check the pupil ids in each class and see how the pupils in a class or an individual pupil interacted with the worked examples.

Thirdly, the student interface of WEAVE is a very close but slightly improved version of the student interface of IWE, and has been able to communicate the worked examples effectively. Furthermore, it provides a means for identifying pupils in a way such that they are identifiable by their teacher but unidentifiable by authors of worked examples and researchers in Computing Science education.

7.1.2.2 Non-Functional Requirements

The majority of the non-functional requirements for the system have been satisfied. The results from the usability evaluation conducted for WEAVE clearly indicate that the prototype is usable, although a number of valuable improvements have been suggested. Tutorial and help guide the users of the system at the major parts of interaction. Shortcuts for transitions between steps are present in the system, although their presence might not be obvious to the users as shown through the evaluation. The panels and the explanation of worked examples are resizable for better control on the system.

Despite the satisfaction of the majority of the non-functional requirements, there is ground for improvement in this area as the results from the heuristics and teacher evaluations indicate. Most importantly, the cross-browser compatibility of the system needs to be improved before WEAVE is deployed in schools. Even though this is not crucial for the functionality of the system, it would improve the user experience.

7.1.3 Current Status of the Project

The prototype is fully functioning and it has been spread around schools so that teachers and pupils have the opportunity to familiarise themselves with the system. They are encouraged to express their opinions about it as well as any ideas for new features that could be beneficial to them.

The code for the system is left in a maintainable state. The methods have comments describing what they do. The separation of concerns imposed by the Model-Template-View design pattern used in Django has been followed. This means that any programmer that has some experience in Django will know where to add code to satisfy new requirements for the system. Further guidance is provided in readme.txt files in the code directories.

7.2 Future Work

This project has been very ambitious. Due to the time constraints, many of the ideas for possible features have not been included to the requirements of this project. These ideas are briefly discussed in this section and will be shortly addressed.

7.2.1 Improvements to the Current Interface

Any issues found in the Usability Evaluation will be addressed in the next version of WEAVE. Most importantly, the problems with the cross-browser compatibility need to be resolved in this version.

7.2.2 Web-Based Author Interface

The next major step for this project is to provide a web-based interface for authors of worked examples. Anonymity of these authors is still under consideration. On one hand, the same privacy issues as with the other users of WEAVE are applicable for authors as well. On the other hand, it could be beneficial to be able to contact them. Another discussion point for this interface is the moderation of the content produced by authors due to the consideration that this content will be visible by pupils.

The author interface will have usage viewing mechanisms similar to the teachers interface, although more in line with the analysis mechanisms outlined in Songs thesis.

7.2.3 Worked Examples Tagging Mechanism

If the system is widely used, the number of worked examples will quickly become unmanageable as a single list. At this point, various additions will be required to manage the complexity:

- tagging of worked examples to enable effective searching by pupil or teacher
- allowing the teacher to attach particular worked examples to particular classes, so that pupils could be focussed onto just the worked examples for them - for example in a My Class Worked Examples section.
- where a teacher can see that a particular worked example is useful for a particular pupil, he/she could add that Worked Example to a My Worked Examples list for that pupil.
- although pupils might have teacher-specified worked examples at either class or individual levels, they could be permitted to search the whole database of worked examples and add any they want to study further to their My Worked Examples list.

7.2.4 Comparison between Time Spent and Expected Time

Songs thesis demonstrated a way of viewing a student interaction with a particular worked example by comparing the times spent at each step with a sequence of times predicted by the worked example author. These predictions were made on the basis of the authors deep knowledge of their own worked example and on an estimation of the how challenging each step would be to a typical learner. Additional interfaces could be added that would allow:

- the author to enter predicted times that a pupil would spend on each step if they were taking the worked example seriously and were at the right developmental stage for the example (i.e. not overly challenged, or way beyond this level already.)
- overlaying of the predicted times onto the average time / student time graphs.

This feature would allow a teacher to very quickly see whether the class as a whole, or individual pupils, were moving faster or slower than expected. If the whole class is significantly slower on a particular step than expected, the teacher will be aware that there is some issue around that point in the worked example. If particular pupils are slower, then some individual support is indicated.

7.3 Personal Development and Reflection

This Level 4 project has been a great opportunity for me. The topic of the project- improving the teaching techniques in Computing Science- has been of particular interest to me for a long time. Coming to the University of Glasgow with absolutely no experience in Computing Science, I have encountered many difficulties typical for novices in Computing Science due to this lack of experience. To me, being part of developing a tool that could reduce the difficulties of other novices is an honour.

During the development of WEAVE, discussions with the project supervisor and, on some meetings, a secondary school teacher gradually built up the required features for the system. These meetings, however, rarely involved discussion on how to implement these features. One of the major personal achievements for me resulting from this project is the ability to analyse different possible solutions and to choose a good option.

Furthermore, despite having experience in Python and Django, I had never used them to build anything as large as this system before. I feel a lot more confident in the technologies used for this project. Building WEAVE helped me to reach a level of expertise satisfactory to become a tutor in the modules involving Django at the University- WAD2 and ITECH.

In addition to the technical skills gained during the last six months, I have hugely developed my communication skills. The most courses at the University develop mainly students technical skills. However, going in industry will most certainly involve giving talks and writing reports. At the beginning of the project, I was not feeling comfortable writing or giving presentations in front of professionals. Receiving constant feedback from the project supervisor helped me hugely improve myself in this area.

Last but not least, this project was a great opportunity for me to make a self-assessment of the knowledge and skills I have been building up during the four years of being a student at the University of Glasgow. I am happy to notice that I am capable of working on large projects myself and satisfying clients requirements. What is left for me to develop is to find a balance between being too self-critical and satisfying myself with an acceptable solution.

7.4 Concluding Remarks

For the past six months most of my time and effort have been concentrated on this project. Looking at the outcome, I feel that they were well spent. The product is a well-accepted and easily deployable system for viewing worked examples in schools and for enabling teachers to better understand the difficulties of their students. Many good software engineering practices have been followed to make the system maintainable and to enhance its evolution.

Despite this advance, the system has a long way to go until it can become revolutionary in the techniques used for teaching Computing Science in the classroom. If this prototype is successful, WEAVE is just the beginning of a much larger system. I am looking forward to being part of the development of this system. For now, though, this is a project I am very satisfied with.

Appendices

Appendix A

Screenshots

This appendix contains screenshots for the different pages in WEAVE.

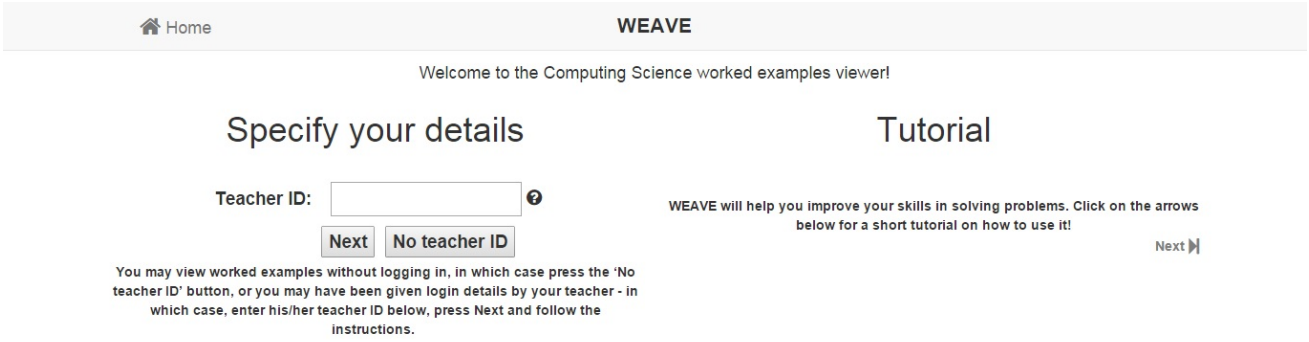


Figure A.1: A screenshot of the main page for the student interface of WEAVE when the pupil has not authenticated themselves.

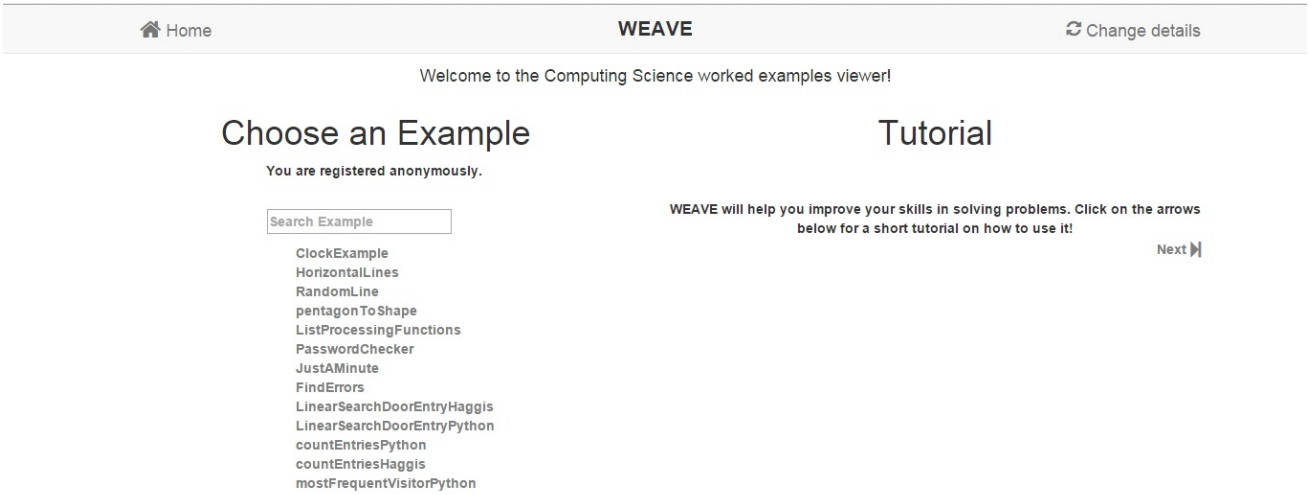


Figure A.2: A screenshot of the main page for the student interface of WEAVE when the pupil has authenticated themselves.

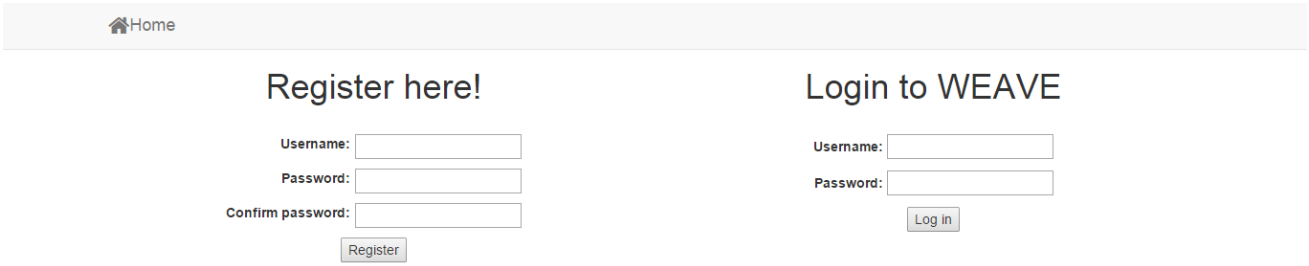


Figure A.3: A screenshot of the main page for the teacher interface of WEAVE for non-logged in teachers.

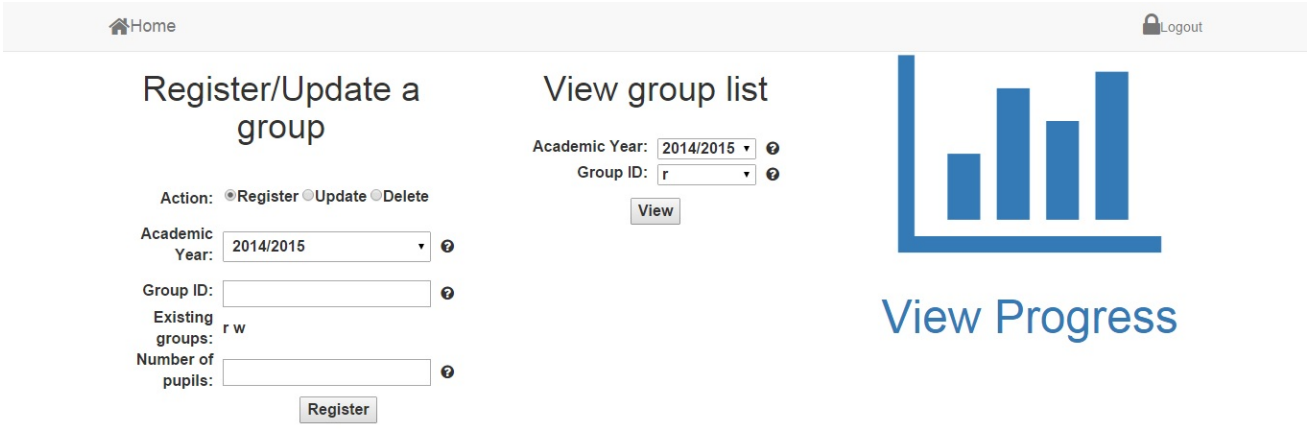


Figure A.4: A screenshot of the main page for the teacher interface of WEAVE for logged in teachers.

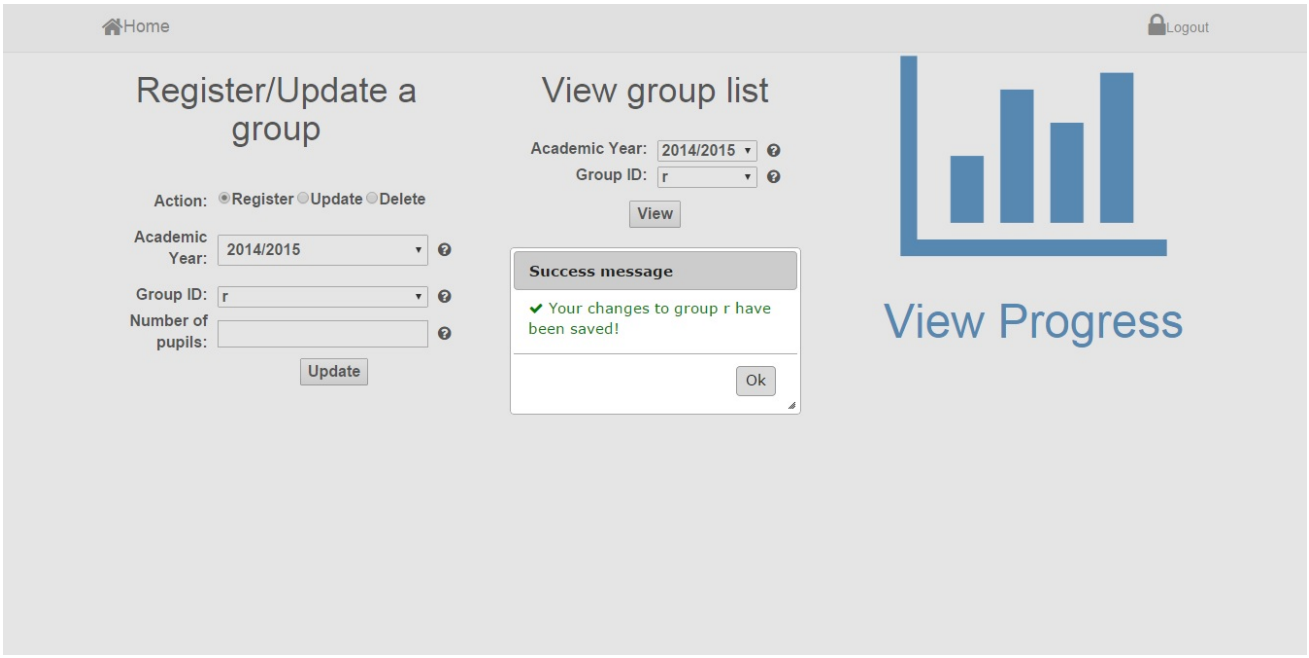


Figure A.5: A screenshot showing a message informing the teacher that their group was successfully saved.

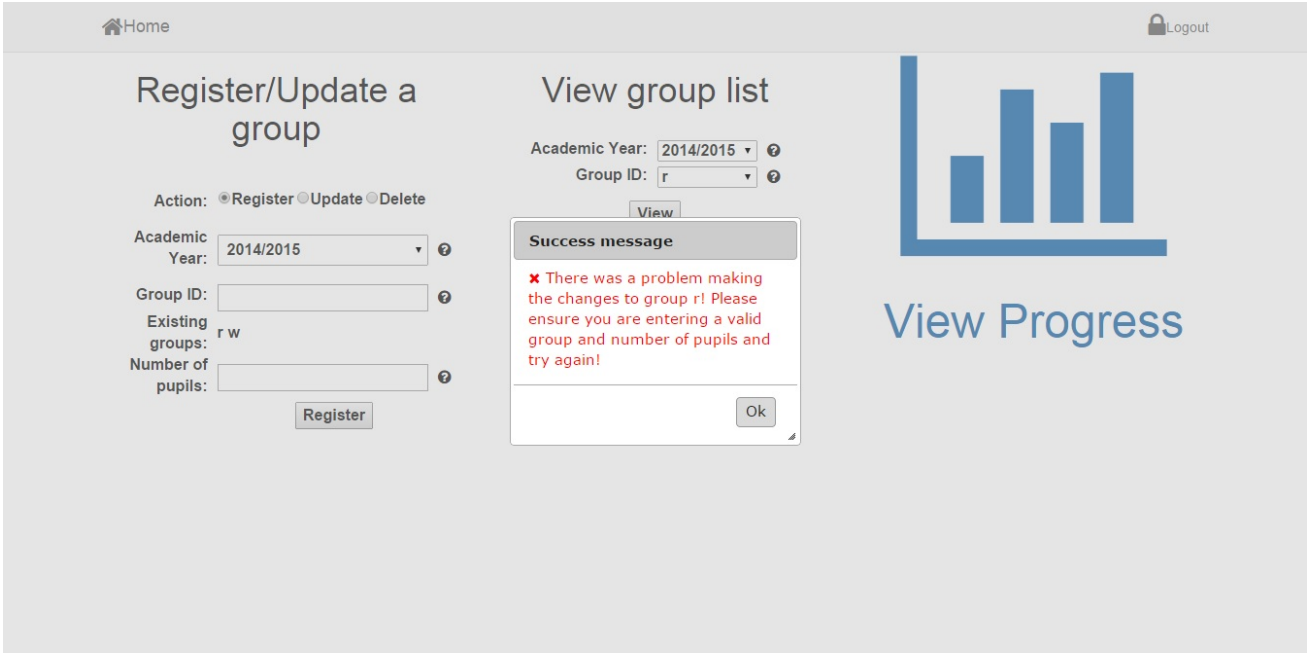


Figure A.6: A screenshot showing a message informing the teacher that their group was not successfully saved.

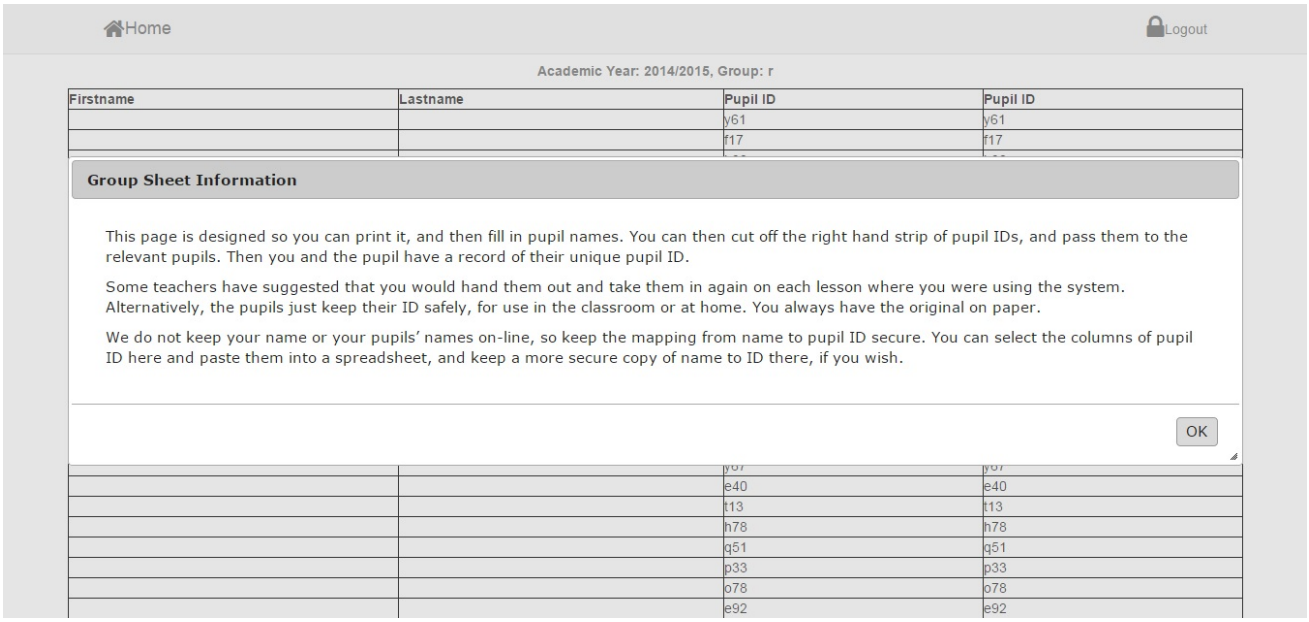


Figure A.7: A screenshot showing information for the intended use of the group sheet.

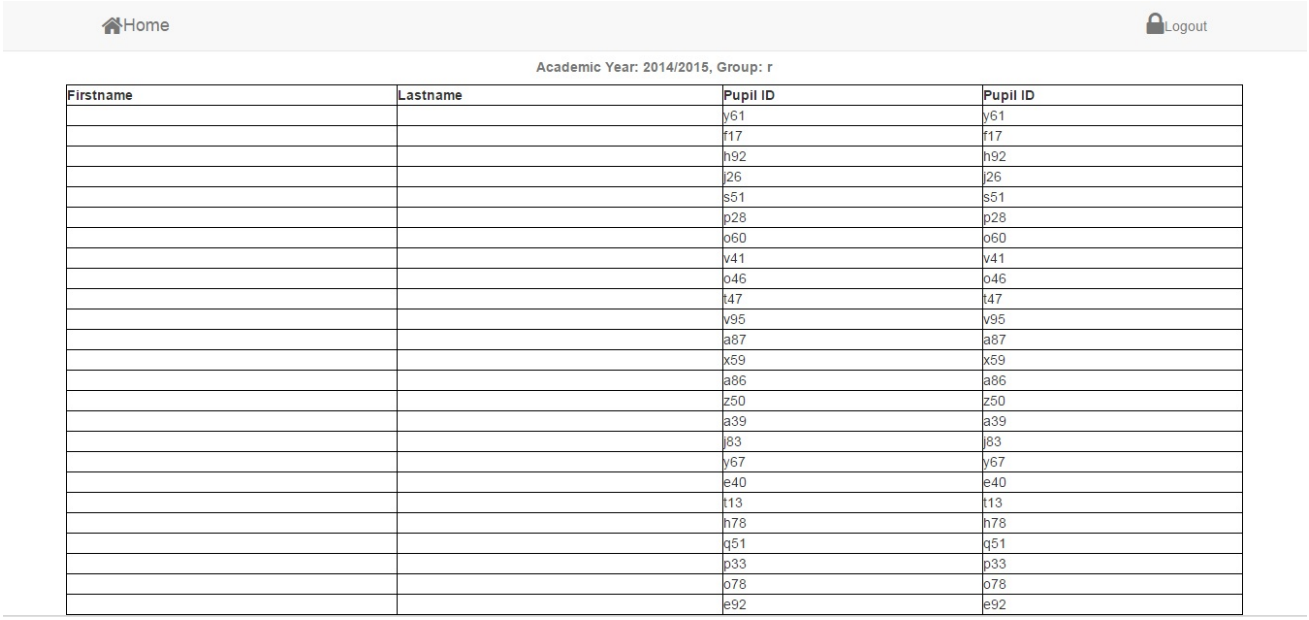


Figure A.8: A screenshot showing the group sheet for a selected group.

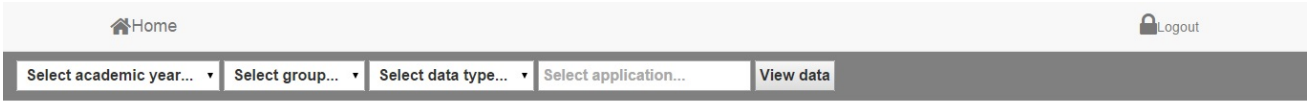


Figure A.9: A screenshot of the progress page in the teacher interface of WEAVE.

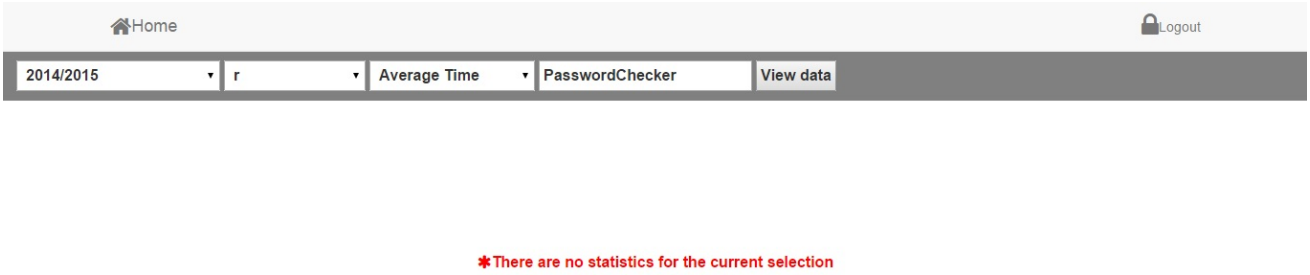


Figure A.10: A screenshot of the progress page in the teacher interface of WEAVE for a selection with no statistics.

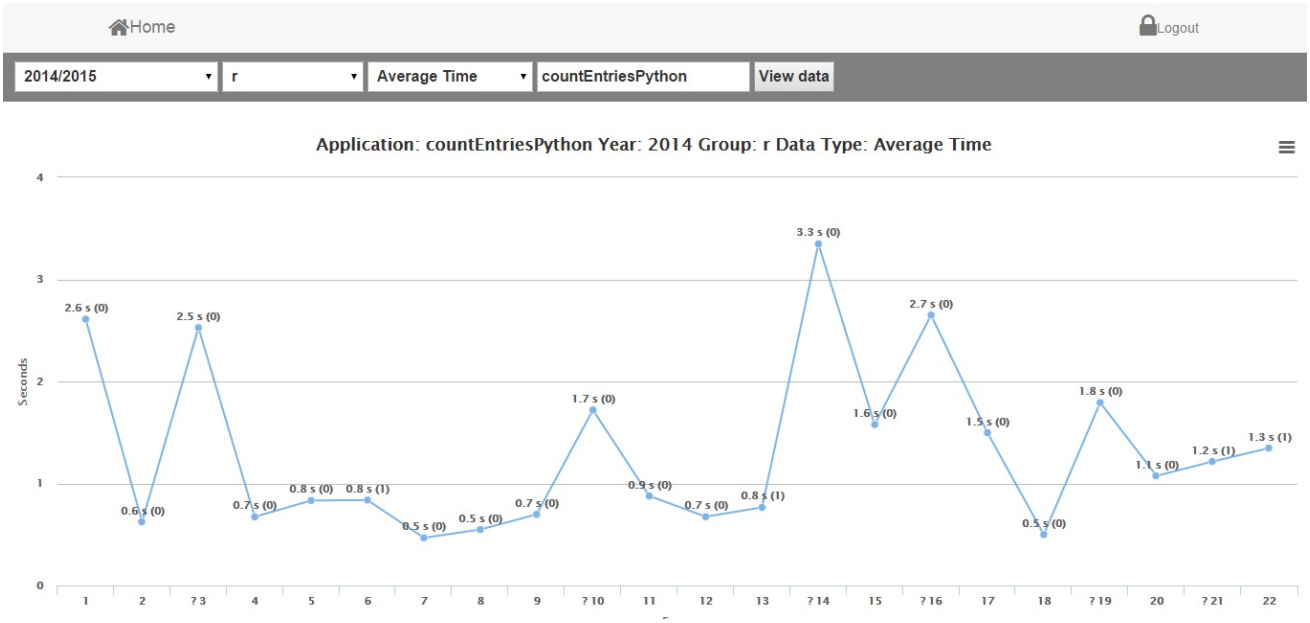


Figure A.11: A screenshot of the progress page in the teacher interface of WEAVE for a selection with statistics.

Appendix B

Consent Form

This is the consent form shown to teachers upon their registration with the system. If they sign this form, the data generated by their pupils will be available to Computing Science researchers for analysis.

Consent form

Information Sheet: Testing the extent to which a new educational tool fits the current educational process in schools The tool

A new web-based application has been developed as a Glasgow University project. It presents worked examples to pupils that allow them to follow approaches to solving problems step by step where a detailed explanation of the way of thinking involved at each step is provided. This application is offered via the local teacher hubs of the PLAN C project, but is available to any teacher in Scotland. It is to be used in schools as a regular teaching technique.

The application supports two interfaces- a student and a teacher interface.

In the teacher's interface, teachers can register with a username and password. This username should be unique to the teacher's institution, but should not identify it. Teachers are asked for their consent to analyse the data associated with their usernames- note, the data cannot be traced back to the school. They are also able to provide an email address so that they receive summary of the analyses made. This email will not be linked to their school username. Once teachers are registered and logged in, they are able to register groups for their classes, as well as individual pupil ids. Furthermore, teachers can see graphs of how students from their groups used the application.

The student interface has the option for students to enter their school and group ids provided by their teacher, as well as an individual pupil id generated by the teacher's interface on request. None of these is compulsory. Pupils are provided with worked examples to study on-line.

Collection of data

The system will collect pupil usage data for the evaluation period 1-28 February 2015. This will be automatically analysed to determine:

-usage patterns during a single session

-usage patterns by the same pupil across multiple sessions, where this possible.

Note that associations between particular real schools, classes and pupils and the data held in the system are not maintained, neither is any other identifying information.

Data will be retained securely for a five-year period to 1 February 2020.

The evaluation

The aim of this evaluation is to test to what extend a newly developed educational tool can be incorporated into the every-day educational process in schools.

At the start, if you want to, you can register with a username and password. This will give you the opportunity to create groups for your classes together with individual pupil ids and analyse the data generated from your pupils via graphs, which will appear on your profile. You would have to provide your pupils with the school, group and pupil ids you generated for their class. Alternatively, you may decide not to give any ids to your pupils. This, however, would mean that you wouldn't be able to see any usage data.

Pupils should attempt several examples provided in the tool.

At the end of the experiment, if you have submitted your email you will be asked to complete a questionnaire.

All the results will remain in strict confidence. It will be your choice of whether you provide us with the school, group and pupil ids generated at the start of the evaluation so that we are able to analyse usage data by school, class and pupil. Note that even if this consent is given, we cannot identify the real school, class or pupil- we are just able to aggregate the data. We would not ask for any identification on an individual level. If you do not give us the consent to use your username and group and pupil ids in our study, we will disallow analysis of any data connected with them.

Please note that this is evaluating to what extend the tool may be used as a teaching technique in schools in the future, and not you or any of the pupils. You and any other participant may withdraw from the experiment at any time and any information recorded will be discarded.

If you have any further questions regarding this experiment, please contact: Emilia Vulpe 1106723v@student.gla.ac.uk

This study adheres to the BPS ethical guidelines.

Appendix C

Sample Group Sheet Printout

This is a sample printout for a group sheet. Teachers are encouraged to fill in the names of their pupils by hand, cut one of the columns for the pupil ids, cut the individual ids, and give them to each of their pupils.

Academic Year: 2014/2015, Group: r

Firstname	Lastname	Pupil ID	Pupil ID
		y61	y61
		f17	f17
		h92	h92
		j26	j26
		s51	s51
		p28	p28
		o60	o60
		v41	v41
		o46	o46
		t47	t47
		v95	v95
		a87	a87
		x59	x59
		a86	a86
		z50	z50
		a39	a39
		j83	j83
		y67	y67
		e40	e40
		t13	t13
		h78	h78
		q51	q51
		p33	p33
		o78	o78
		e92	e92

Appendix D

Usage Graphs

This Appendix contains screenshots for the different graphs supported by the teacher interface.

D.1 Average/Pupil Time Graph

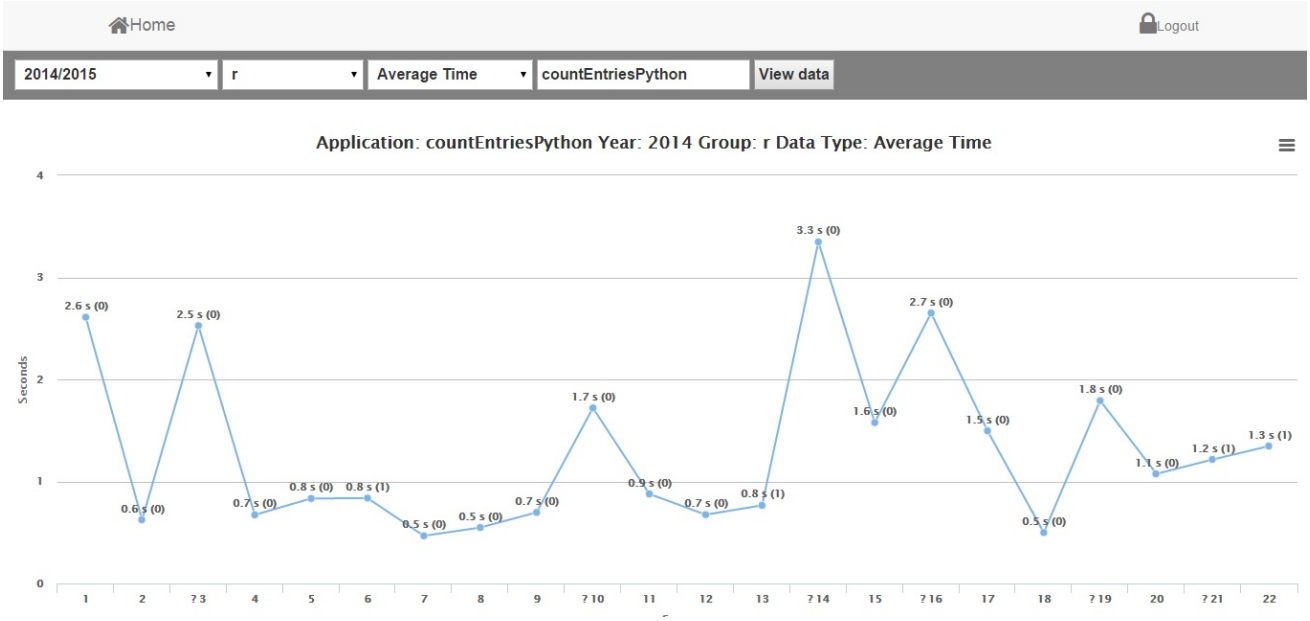


Figure D.1: A screenshot of the plain graph showing the time pupils in a group or an individual pupil spent on the steps of an example.

D.2 Class Steps

D.3 Class Summary

D.4 Pupil Answers

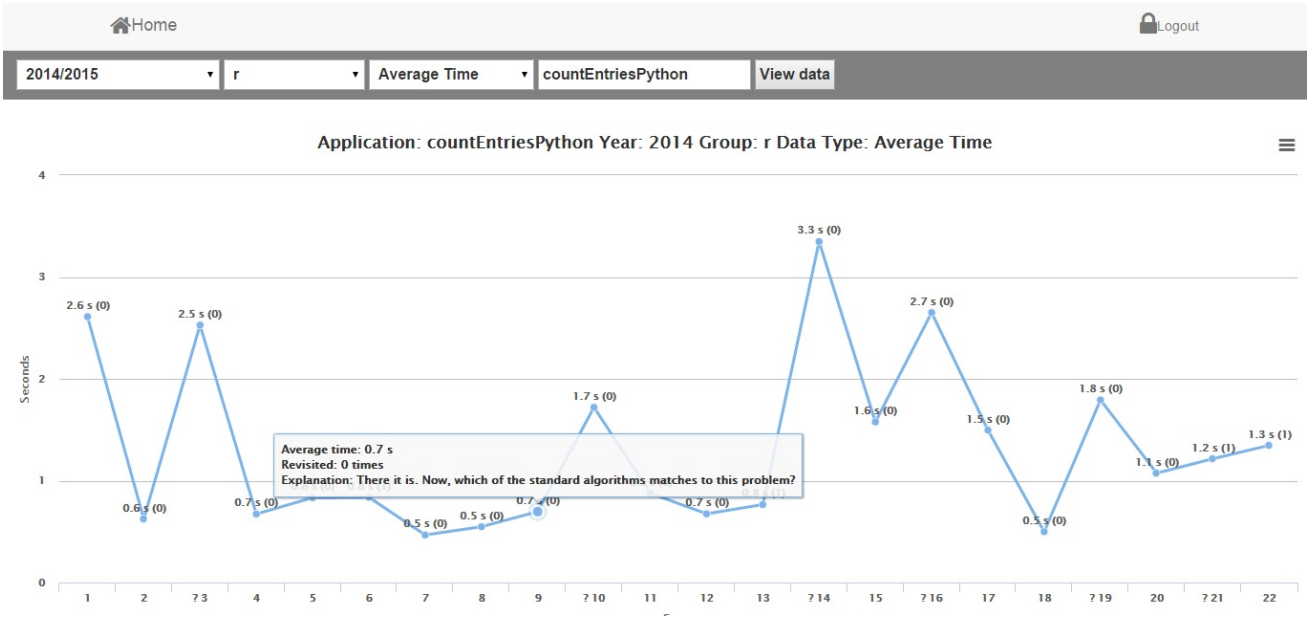


Figure D.2: A screenshot of the graph showing the time pupils in a group or an individual pupil spent on the steps of an example, and the explanation shown on hovering over the step point.

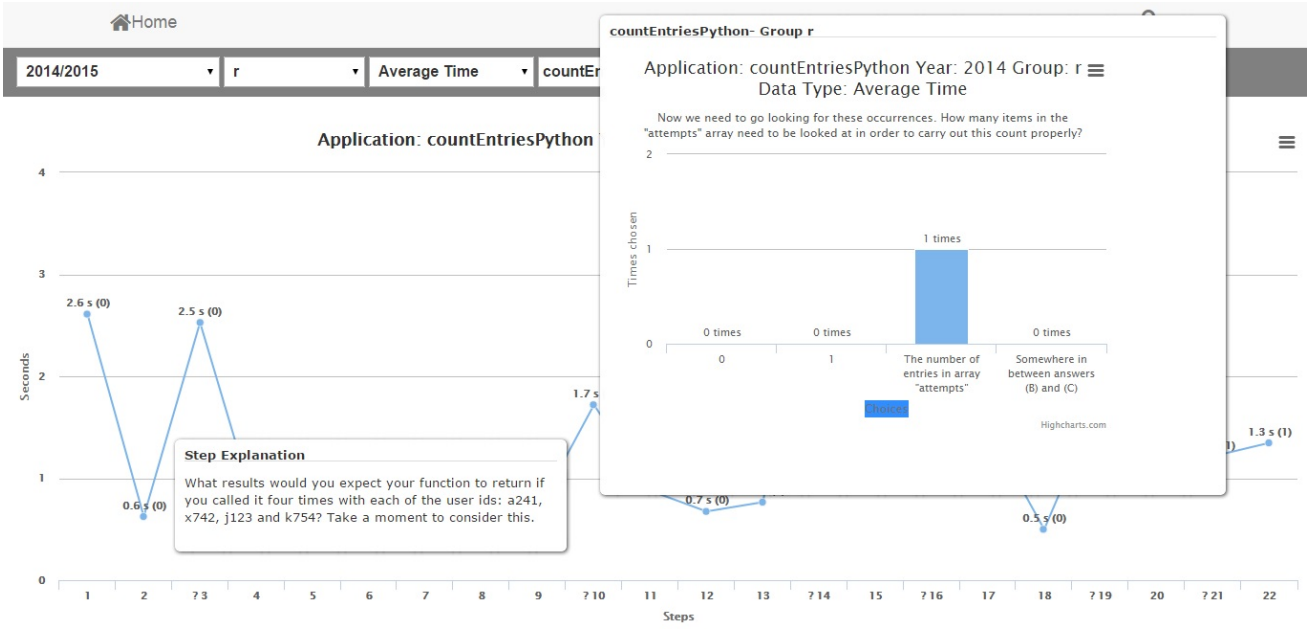


Figure D.3: A screenshot of the graph showing the time pupils in a group or an individual pupil spent on the steps of an example. A step without a question (on the left) and a step with a question (on the right) are clicked over with the mouse.

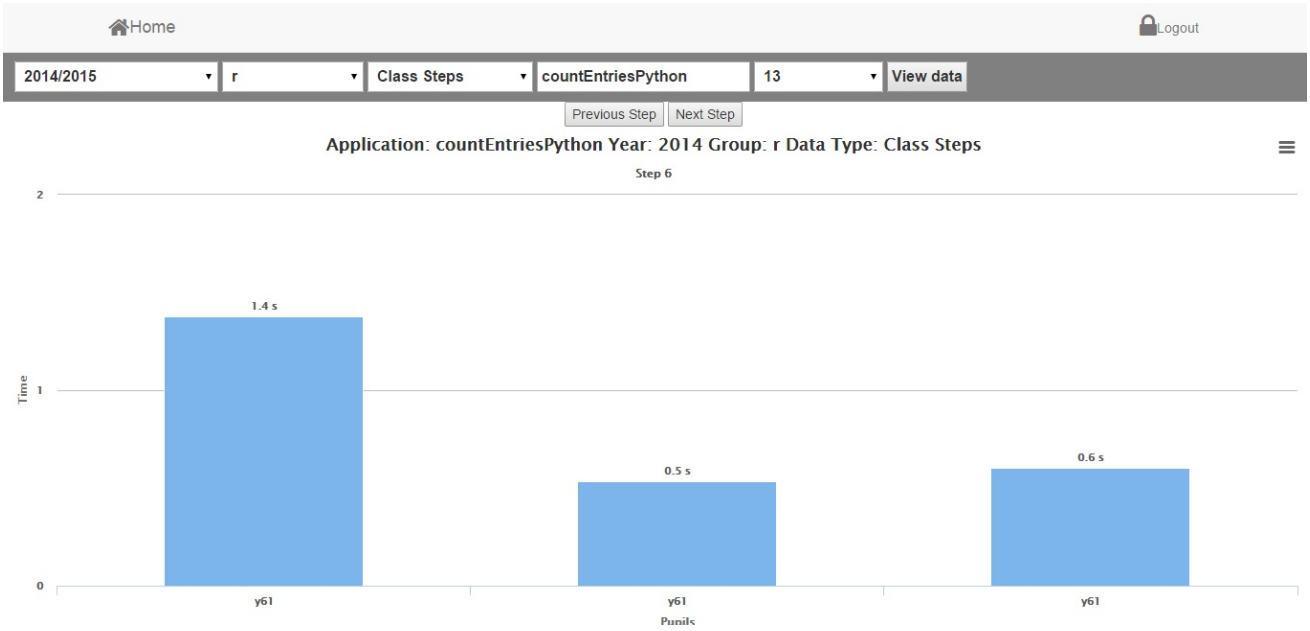
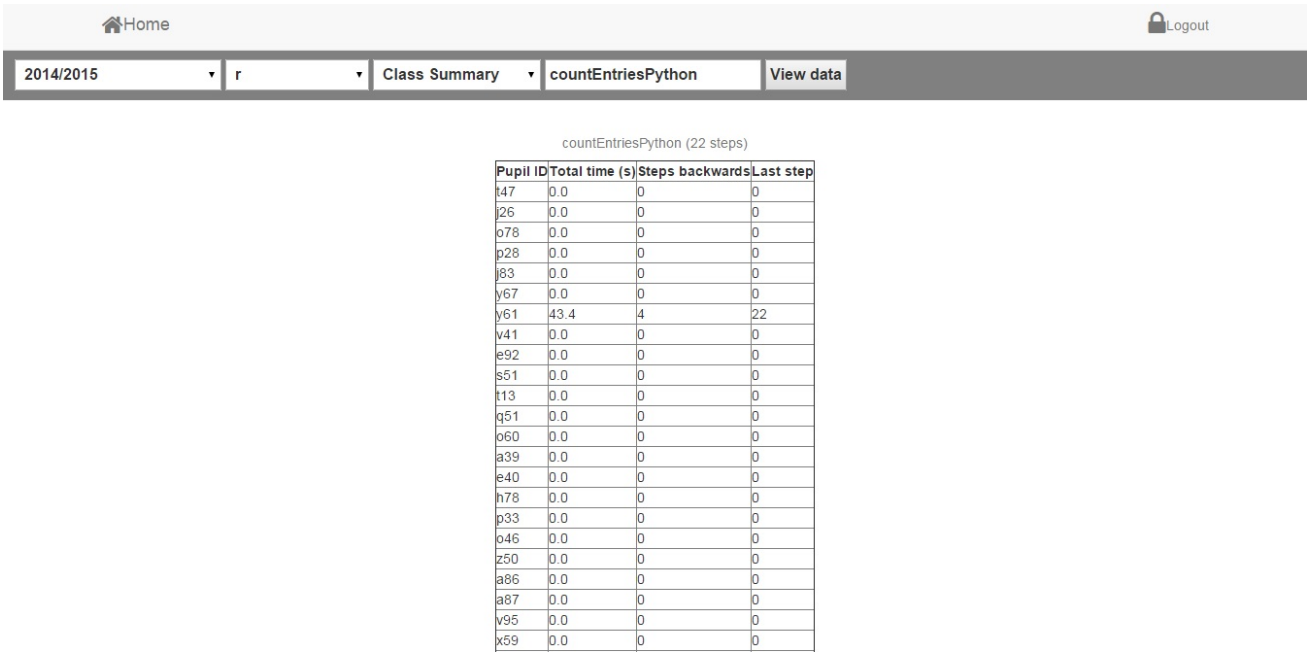


Figure D.4: A screenshot of the Class Steps graph.



Pupil ID	Total time (s)	Steps backwards	Last step
t47	0.0	0	0
j26	0.0	0	0
o78	0.0	0	0
p28	0.0	0	0
j83	0.0	0	0
y67	0.0	0	0
y61	43.4	4	22
v41	0.0	0	0
e92	0.0	0	0
s51	0.0	0	0
t13	0.0	0	0
q51	0.0	0	0
o60	0.0	0	0
a39	0.0	0	0
e40	0.0	0	0
h78	0.0	0	0
p33	0.0	0	0
o46	0.0	0	0
z50	0.0	0	0
a86	0.0	0	0
a87	0.0	0	0
v95	0.0	0	0
x59	0.0	0	0

Figure D.5: A screenshot of the Class Summary table for an example.

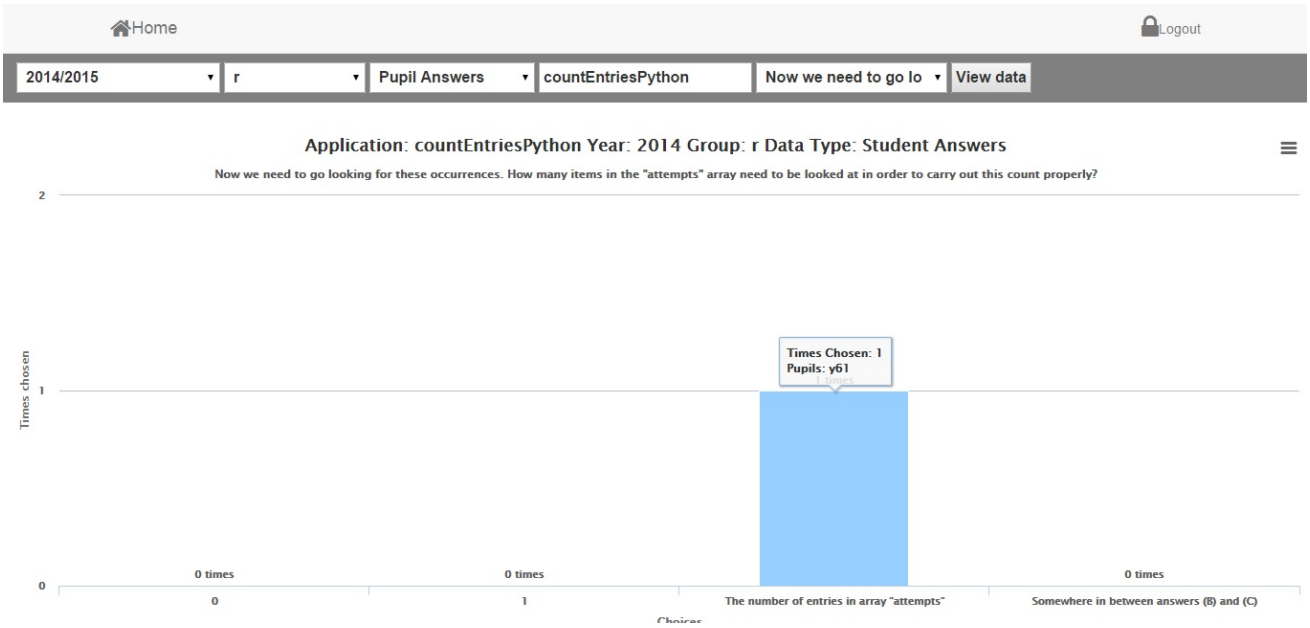


Figure D.6: A screenshot of the graph showing the answers selected by pupils for a selected question of an example.

Appendix E

Evaluation Materials and Responses

E.1 Questions

This Appendix contains the questionnaires used for the evaluation of WEAVE.

E.1.1 Information sheet and a questionnaire for the usability evaluation with participants with no experience in Computing Science

Evaluation Information Sheet

In this session, you will be working with an online worked examples viewer (WEAVE). The main characteristic of worked examples is that their text is revealed gradually as you work through different steps of the example.

Context

WEAVE is intended to be a new tool which can be used to enhance the teaching process of Computing Science in schools across the UK.

Teachers

WEAVE enables teachers to make use of worked examples created by them or by their colleagues. They will be able to work with their students on selected examples in the classroom or set them as homework. The benefit WEAVE brings to them is that they can monitor how their students interact with these worked examples. In order to do so, teachers are encouraged to organise their classes into groups. After selecting the number of students for each group, the tool will generate IDs for each student. The teacher is responsible for passing their own username, which serves as a teacher ID, the group ID and the student ID to the students they want to monitor usage data for. WEAVE will enable them to see such data at a class and at an individual level. The data is visualised as graphs and tables for easier analysis.

Students

For students WEAVE serves as a worked examples viewer. They need to identify themselves by the teacher ID, group ID and student ID provided by their teacher in order to be able to view examples. The system also allows access to examples to individuals who are not assigned to any group- they need to identify themselves as anonymous users. When they choose an example to work on, students go through the different steps of the example and read the explanation for the step they are on. Some steps involve questions to check student's knowledge.

The evaluation

Instructions on what you are supposed to achieve during interacting with WEAVE are provided below. These instructions, however, are minimal because one of the aspects of the evaluation is to evaluate how self-explanatory WEAVE is.

The evaluation consists of three stages.

Stage 1

In the first stage you will act as teacher. You need to imagine that you want to monitor the progress for a class as well as the progress of one of your students. Make sure that you create a group for this purpose.

Stage 2

In the second stage you will act as a student. You have received a teacher ID, group ID and a student ID and you need to choose one or more examples to work on. Note that you don't need to understand the example(s) as you go through the steps so you should not worry if you don't. The focus of the evaluation is the user interface.

Stage 3

You act as a teacher again. You want to check how your class worked with the example(s) used in the previous stage. You also want to check how the particular student (the one you chose for your second stage of the evaluation) has worked with this example(s).

Comments

Please note that this is evaluation on how effective the application is in achieving its goals and so you should identify any potential issues preventing you from working with WEAVE in an effective manner. It is not evaluating you or your skills. You may withdraw from the evaluation at any time and any information recorded will be discarded.

Questions

Usability Evaluation

1. What is your overall impression on the application?

2. Is there anything in particular about which you have a strong opinion - either good or bad?

3. Did the prototype provide enough guidance and help on what you needed to do at each stage?

4. Do you think you would need special training on how to use WEAVE? If yes, can you specify what parts of the application should be the focus of such training?

5. Were there any parts of interaction when you were confused of unsure what you need to do next? If yes:

    5.1 How did you act to find out what you needed to do?

5.2. How difficult was it to find out what you needed to do?

5.3. How certain you felt that your actions are appropriate?

6. Is there anything that could improve your experience?

7. Do you have any final thoughts or opinion about WEAVE?

E.1.2 Questionnaire for the usability and the practical evaluation with teachers

Your use of WEAVE

Please tick which of the following uses you have made of WEAVE (as many as apply)

1. ____ I viewed worked examples myself using the pupil interface
2. ____ I looked at the teacher interface
3. ____ I used WEAVE in one or more classes with pupils
4. ____ I used WEAVE with pupils and then used the teacher interface to explore their progress

Usability

When answering the following, note that each question has space for two answers, one for the teacher interface and one for the pupil interface. If you didn't use one or other interface, just leave that part unanswered.

1. I think that I would like to use this system frequently

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	5

---

2. I found the system unnecessarily complex

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

3. I thought the system was easy to use

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree			Strongly agree
1	2	3	4

---

4. I think that I would need the support of a technical person to be able to use this system

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree			Strongly agree
1	2	3	4

---

5. I found the various functions in this system were well integrated

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree			Strongly agree
1	2	3	4

---

6. I thought there was too much inconsistency in this system

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

7. I would imagine that most people would learn to use this system very quickly

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

8. I found the system very cumbersome to use

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

9. I felt very confident using the system

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree			Strongly agree
1	2	3	4

---

10. I needed to learn a lot of things before I could get going with this system

Pupil Interface

Strongly disagree				Strongly agree
1	2	3	4	5

Teacher Interface

Strongly disagree			Strongly agree
1	2	3	4

Please comment on aspects of the **teacher** interface that didn't work properly or could be improved.

Please comment on aspects of the **pupil** interface that didn't work properly or could be improved.

Fitness for teaching and learning

If there were more worked examples in WEAVE would you use it in your classes regularly? Please explain your answer.

Presentation of pupils' usage data

How easy was it to understand what information did the different graphs/tables show?

Very difficult				Very easy
1	2	3	4	5

How useful did you find the graphs/tables in understanding how your pupils worked with the examples?

Not useful at all				Very useful
1	2	3	4	5

Is there any other information that the system does not provide but would help you to better understand the difficulties of your pupils?

Deploying WEAVE in schools

How easy do you think would be to deploy WEAVE in your school?

Very difficult				Very easy
1	2	3	4	5

Logistics

Are there any logistical problems with using WEAVE in your classroom? Possible issues are: website blocking, browser incompatibilities, machine limitations (speed, screen size etc.)

Other comments

Do you have any other comments to make about the system? For example, most of these questions ask about problems… in addition, what do you like about the system?!

E.1.3 Questionnaire for the heuristics evaluation with Django experts

Heuristics Evaluation Questions

1. Visibility of system status

- Did you receive enough feedback of the system's status?

2. Match between system and real world

- Is the terminology used familiar to you?
- Does the interface present information in a simple, natural and logical order?
- Can you easily understand the use of metaphors?

3. User control and freedom

- Do you feel in control with the system?
- Is there anything that you think restricted you to complete a task?
  Do you think that the facilities provided to return to the top level at any stage are appropriate?
- Do you think that the facilities provided to recover from an error were appropriate?

4. Consistency and standards

- Do you think that the terminology and the controls are consistent throughout the system?
- Is the look and feel of the system consistent?

5. Error prevention

- Are the input options chosen suitably to prevent errors?
- Recognition rather than recall
- *Were you able to access help and instructions easily when needed?*
- *Did you* find the relationship between controls and their actions obvious?

6. Flexibility and efficiency of use

- Does the website provide any shortcuts for improving interactions?
- Does the website interface guide you how to use it sufficiently?
- Did you feel some of the registrations were unnecessary?

7. Aesthetic and minimalist design

- Did you find the design simple, easy to learn and intuitive?
- Do the interfaces have any information which you found irrelevant and distracting?
- Are the different controls of the system clearly labelled and obvious?

8. Help users recover from errors

- Did you find any error messages useful to help you recover from the error?
- Was it clear by the error messages what you have done wrong?

9. Help and documentation

- Is sufficient help provided?
- Are the help messages clear enough for you to understand them?

10. Navigation

- Did you feel at any point that navigation in the website was inappropriate or unnecessary?
- Was it clear to you what page you are on and where can you go next?

11. Structure of information

- Did you feel that related pieces of information are clustered together?
- Is the size of the different elements appropriate in respect to the size of the screen and their importance?

12. Physical constraints

- Is the distance between different elements and their size appropriate?

E.1.4 SUS Questionnaire

SUS Questionnaire

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently					
	1	2	3	4	5
2. I found the system unnecessarily complex					
	1	2	3	4	5
3. I thought the system was easy to use					
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system					
	1	2	3	4	5
5. I found the various functions in this system were well integrated					
	1	2	3	4	5
6. I thought there was too much inconsistency in this system					
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly					
	1	2	3	4	5
8. I found the system very cumbersome to use					
	1	2	3	4	5
9. I felt very confident using the system					
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system					
	1	2	3	4	5

E.2 Usability Evaluation

This appendix contains copies of the evaluation forms that were filled out by participants. They are split into two main sections:

- usability evaluation with participants with no experience in Computing Science; and
- usability evaluation with school teachers in Computing Science.

E.2.1 Participants With No Prior Experience

E.2.1.1 Participant 1

This subsection includes:

- participant 1's question responses; and
- participant 1's student and teacher interface SUS responses.

Questions

Usability Evaluation

1. What is your overall impression on the application?

The overall impression of the application is good – it seems to cover all the bases for the requirements that I gathered from the description provided to me for the evaluation. The interface is simple enough to understand and pretty intuitive. I was able to complete all of the tasks asked of me. So yes, I think overall it is a good application with promise!

2. Is there anything in particular about which you have a strong opinion - either good or bad?

The interface is good, I have a good opinion of the interface. I can't think of anything bad which I have strong opinions of! I think the major issue is the potentially confusing way in which graphs are displayed – the numbers are a bit confusing at first.

3. Did the prototype provide enough guidance and help on what you needed to do at each stage?

Pretty much. I understood the major points, such as how to log in, find student IDs for my group, and the like. This all seemed fine – the buttons/links were where I expected them to be. I was able to successfully complete the tasks without major issues.

My only problem was identifying on the average time charts what the numbers meant next to the graph points. I had no idea; then I found out when you hover the mouse over a point, you get a popup bubble with some descriptions of the particular step. I found that numbers in the bubble (with a text description accompanying them) matched the numbers in the brackets I had no idea about – so I deduced that those were the same values. It might be good to have a little popup when a graph loads telling the user what the values mean.

4. Do you think you would need special training on how to use WEAVE? If yes, can you specify what parts of the application should be the focus of such training?

*For students, no. I think students would be able to pick up and use the system pretty easily. Nice one! For teachers, *maybe*. I am not totally sure; the actions the teachers have to do is a little more complex since they have administrative work to carry out here. What I would say is that a two or three page guide on how to set up the system from a teacher's perspective would be very handy for them.*

5. Were there any parts of interaction when you were confused of unsure what you need to do next? If yes:

Yes – but only one (graphs, trying to figure them out).

5.1 How did you act to find out what you needed to do?

I was totally confused at first. After moving the mouse around the screen, I realised that when you hovered over the points on the graph, you got a little popup telling you more about each point. This helped me identify the values on the graph.

5.2. How difficult was it to find out what you needed to do?

Pretty difficult because I had no guidance. Some help would have been useful here.

5.3. How certain you felt that your actions are appropriate?

Very – because the numbers always matched. So I am confident that I figured out the correct meaning behind each of the values.

6. Is there anything that could improve your experience?

I think the data reporting section could be improved a little to make it easier to use and digest. I am aware the previous incarnation of the software had no such functionality; so I feel that what is present is a great first attempt. However, some refinement would go a long way. Some of the graphs don't seem particularly suitable to be presented as graphs. Maybe just tables, instead?

However, the student interface seems to have been well executed. I have no comments on this side of things regarding how that can be improved. Well done!

7. Do you have any final thoughts or opinion about WEAVE?

For a Level 4 project, I think WEAVE clearly demonstrates what is required for such a project, plus a lot more. At first, I thought the application was rather simplistic. When I however started to think about the mechanics that must have been implemented to develop such a product, it became much clearer to me that this is much more than a simple web app. It is sophisticated, with great logging functionality, and I am sure that in the years to come, WEAVE and any products spun off from it will aid in the teaching of complex programming activities to students.

SUS Questionnaire
Student Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently				✔	
	1	2	3	4	5
2. I found the system unnecessarily complex	✔				
	1	2	3	4	5
3. I thought the system was easy to use					✔
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated				✔	
	1	2	3	4	5
6. I thought there was too much inconsistency in this system			✔		
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly					✔
	1	2	3	4	5
8. I found the system very cumbersome to use		✔			
	1	2	3	4	5
9. I felt very confident using the system					✔
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system		✔			
	1	2	3	4	5

SUS Questionnaire
Teacher Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently			✔		
	1	2	3	4	5
2. I found the system unnecessarily complex		✔			
	1	2	3	4	5
3. I thought the system was easy to use				✔	
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated				✔	
	1	2	3	4	5
6. I thought there was too much inconsistency in this system			✔		
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly					✔
	1	2	3	4	5
8. I found the system very cumbersome to use		✔			
	1	2	3	4	5
9. I felt very confident using the system					✔
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system		✔			
	1	2	3	4	5

E.2.1.2 Participant 2

This subsection includes:

- participant 2's question responses; and
- participant 2's student and teacher interface SUS responses.

Questions

Usability Evaluation

1. **What is your overall impression on the application?**

   It can be a really useful application. A great way to spot the student's efforts and time spent on exercising outside of the classroom.

2. **Is there anything in particular about which you have a strong opinion - either good or bad?**

   The thing in particular that I liked the most is the guidance through the examples, which simultaneously combined with questions, which is a great learning method.

3. **Did the prototype provide enough guidance and help on what you needed to do at each stage?**

   Regarding the examples- Yes. Regarding the usability of the application, I think there should be a tutorial that guides the users through the first use.

4. **Do you think you would need special training on how to use WEAVE? If yes, can you specify what parts of the application should be the focus of such training?**

   I think that special training is definitely not needed, but as I mentioned a quick guidance tutorial will be helpful on the first time a user opens the application.

5. Were there any parts of interaction when you were confused of unsure what you need to do next? If yes:

5.1 How did you act to find out what you needed to do?

5.2. How difficult was it to find out what you needed to do?

5.3. How certain you felt that your actions are appropriate?

No.

6. Is there anything that could improve your experience?

I had a problem with overlapping texts that was probably caused by the version of my browser. I could not see the explanations as they were overlapping with the solution on the screen. If that problem is solved, I have no further suggestions.

7. Do you have any final thoughts or opinion about WEAVE?

Once applied in a school if can definitely help both pupils and teachers. The pupils can benefit from WEAVE through having the guidance, while going through an example, which they would most likely give up on without any help or questions asked, that way when their learning process is supported

and made more interesting, they can invest more of their time and increase their performance. For the teachers it is useful to see how much extra time a pupil will put in and also to see the statistics and results with compared on the time invested. I also think that WEAVE has a great potential of expansion, once it is applied for a group of pupils.

SUS Questionnaire
Student Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently			✔		
	1	2	3	4	5
2. I found the system unnecessarily complex	✔				
	1	2	3	4	5
3. I thought the system was easy to use					✔
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated					✔
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	✔				
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly					✔
	1	2	3	4	5
8. I found the system very cumbersome to use	✔				
	1	2	3	4	5
9. I felt very confident using the system				✔	
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	✔				
	1	2	3	4	5

SUS Questionnaire
Teacher Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently				✔	
	1	2	3	4	5
2. I found the system unnecessarily complex	✔				
	1	2	3	4	5
3. I thought the system was easy to use					✔
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated					✔
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	✔				
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly				✔	
	1	2	3	4	5
8. I found the system very cumbersome to use	✔				
	1	2	3	4	5
9. I felt very confident using the system				✔	
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	✔				
	1	2	3	4	5

E.2.1.3 Participant 3

This subsection includes:

- participant 3's question responses; and
- participant 3's student and teacher interface SUS responses.

Questions

Usability Evaluation

1. What is your overall impression on the application?

Anna: It seems like a useful tool which is not too complicated to master and use easily.

2. Is there anything in particular about which you have a strong opinion - either good or bad?

Anna: It was very easy to use and most of the things were intuitive.

3. Did the prototype provide enough guidance and help on what you needed to do at each stage?

Anna: Yes. With the extra tutorial and self-explanatory functions it was clear what to do at every stage.

4. Do you think you would need special training on how to use WEAVE? If yes, can you specify what parts of the application should be the focus of such training?

Anna: I would think that if an older person (60+) would have had some problems with the program due to lack of experience. I do not see a problem for computer experienced people.

5. Were there any parts of interaction when you were confused of unsure what you need to do next? If yes:

5.1 How did you act to find out what you needed to do?

Anna: I just clicked around a bit.

5.2. How difficult was it to find out what you needed to do?

Anna: Not very hard, the links and the buttons are very clear.

5.3. How certain you felt that your actions are appropriate?

Anna: Quite certain. When I got to where I needed to be that confirmed it.

6. Is there anything that could improve your experience?

Anna: Just small things(a line in one place, use of same words for tutorial and the actual links). Nothing major about the program and its functionality.

7. Do you have any final thoughts or opinion about WEAVE?

Anna: As a teacher teaching out e.g python, the examples shown step by step would be good. Also being able to monitor my students' activities and where they spent most time would be very helpful.

SUS Questionnaire
Student Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently					✔
	1	2	3	4	5
2. I found the system unnecessarily complex		✔			
	1	2	3	4	5
3. I thought the system was easy to use				✔	
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated					✔
	1	2	3	4	5
6. I thought there was too much inconsistency in this system		✔			
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly				✔	
	1	2	3	4	5
8. I found the system very cumbersome to use		✔			
	1	2	3	4	5
9. I felt very confident using the system				✔	
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system		✔			
	1	2	3	4	5

SUS Questionnaire
Teacher Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently					✔
	1	2	3	4	5
2. I found the system unnecessarily complex		✔			
	1	2	3	4	5
3. I thought the system was easy to use				✔	
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated				✔	
	1	2	3	4	5
6. I thought there was too much inconsistency in this system		✔			
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly			✔		
	1	2	3	4	5
8. I found the system very cumbersome to use			✔		
	1	2	3	4	5
9. I felt very confident using the system			✔		
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system			✔		
	1	2	3	4	5

E.2.1.4 Participant 4

This subsection includes:

- participant 4's question responses; and
- participant 4's student and teacher interface SUS responses.

Questions

Usability Evaluation

1. What is your overall impression on the application?

I think it is very useful for teachers to give tasks to students – for students to complete them and for teachers to check them – this cycle is very easily performed by this application.

2. Is there anything in particular about which you have a strong opinion - either good or bad?

Good: usefulness of the app as well as the easiness of working with it;
Bad: the design is poor – it lacks colours, a little bit "dry"

3. Did the prototype provide enough guidance and help on what you needed to do at each stage?

5/6 it can be completed with a just a few more guiding instructions.

4. Do you think you would need special training on how to use WEAVE? If yes, can you specify what parts of the application should be the focus of such training?

No, the app is very easy to use.

5. Were there any parts of interaction when you were confused of unsure what you need to do next? If yes:

5.1 How did you act to find out what you needed to do?

Yes, maybe a little bit more initial instructions would help.

5.2. How difficult was it to find out what you needed to do?

Not applicable.

5.3. How certain you felt that your actions are appropriate?

Not applicable.

6. Is there anything that could improve your experience?

More colour to the app would be improve the UX. Maybe also include a chance for the students to comment on their tasks (or on specific parts of the task)

7. Do you have any final thoughts or opinion about WEAVE?

It's useful both for students and teachers, it easy to use and it has the potential to be improved further.

SUS Questionnaire
Student Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently				✔	
	1	2	3	4	5
2. I found the system unnecessarily complex	✔				
	1	2	3	4	5
3. I thought the system was easy to use					✔
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated				✔	
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	✔				
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly					✔
	1	2	3	4	5
8. I found the system very cumbersome to use	✔				
	1	2	3	4	5
9. I felt very confident using the system					✔
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	✔				
	1	2	3	4	5

SUS Questionnaire
Teacher Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently				✔	
	1	2	3	4	5
2. I found the system unnecessarily complex	✔				
	1	2	3	4	5
3. I thought the system was easy to use				✔	
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated			✔		
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	✔				
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly					✔
	1	2	3	4	5
8. I found the system very cumbersome to use	✔				
	1	2	3	4	5
9. I felt very confident using the system					✔
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	✔				
	1	2	3	4	5

E.2.1.5 Participant 5

This subsection includes:

- participant 5's question responses; and
- participant 5's student and teacher interface SUS responses.

Questions

Usability Evaluation

1. **What is your overall impression on the application?**

   By reading carefully the instructions – the application seems very easy to work with. Also, it doesn't take much time to register and to start working with.

2. **Is there anything in particular about which you have a strong opinion - either good or bad?**

   The application can be very useful for the learning process of children/ students and teachers not only for the ones studying IT, but also for other subjects.

   The students can practice what they have learned and better understand something they have difficulties with and the teachers can easily evaluate the students' performance, the time they have invested on a certain topic, the level of knowledge of a group/ class and also to evaluate their own methods of teaching, etc.

3. **Did the prototype provide enough guidance and help on what you needed to do at each stage?**

   Yes, the guide describes in the most clear and informative way what is to be done on each of the stages.

4. **Do you think you would need special training on how to use WEAVE? If yes, can you specify what parts of the application should be the focus of such training?**

   No, I don't think that special training on how to use WEAVE is needed.

5. **Were there any parts of interaction when you were confused of unsure what you need to do next?**

   No, I didn't have difficulties with what I have to do next.

   If yes:

   5.1 How did you act to find out what you needed to do?

   5.2. How difficult was it to find out what you needed to do?

   5.3. How certain you felt that your actions are appropriate?

6. Is there anything that could improve your experience?

I think that it will be very useful if the application is made accessible through mobile devices the same way it is through a computer.

7. Do you have any final thoughts or opinion about WEAVE?

I would like to have the opportunity to use the application in my studying process because it will ease the process of practicing and understanding the material taught in the university.

SUS Questionnaire
Student Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently					✔
	1	2	3	4	5
2. I found the system unnecessarily complex	✔				
	1	2	3	4	5
3. I thought the system was easy to use					✔
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated					✔
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	✔				
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly					✔
	1	2	3	4	5
8. I found the system very cumbersome to use	✔				
	1	2	3	4	5
9. I felt very confident using the system				✔	
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	✔				
	1	2	3	4	5

SUS Questionnaire
Teacher Interface

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently					✔
	1	2	3	4	5
2. I found the system unnecessarily complex	✔				
	1	2	3	4	5
3. I thought the system was easy to use					✔
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	✔				
	1	2	3	4	5
5. I found the various functions in this system were well integrated				✔	
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	✔				
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly					✔
	1	2	3	4	5
8. I found the system very cumbersome to use		✔			
	1	2	3	4	5
9. I felt very confident using the system			✔		
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system		✔			
	1	2	3	4	5

E.2.2 Teachers

E.2.2.1 Participant 1

Your use of WEAVE

Please tick which of the following uses you have made of WEAVE (as many as apply)

1. _√__ I viewed worked examples myself using the pupil interface
2. _√__ I looked at the teacher interface
3. ____ I used WEAVE in one or more classes with pupils
4. ____ I used WEAVE with pupils and then used the teacher interface to explore their progress

Usability

The following is the standard SUS usability survey. When answering these, note that each question has space for two answers, one for the teacher interface and one for the pupil interface. If you didn't use one or other interface, just leave that part unanswered.

1. I think that I would like to use this system frequently

Pupil Interface

Strongly disagree				Strongly agree
			√	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		√		
1	2	3	4	5

---

2. I found the system unnecessarily complex

Pupil Interface

Strongly disagree				Strongly agree
	√			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
	√			
1	2	3	4	

3. I thought the system was easy to use

Pupil Interface

Strongly disagree				Strongly agree
		√		
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
				√
1	2	3	4	

---

4. I think that I would need the support of a technical person to be able to use this system

Pupil Interface

Strongly disagree				Strongly agree
√				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
√				
1	2	3	4	

---

5. I found the various functions in this system were well integrated

Pupil Interface

Strongly disagree				Strongly agree
			√	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
			√	
1	2	3	4	

---

6. I thought there was too much inconsistency in this system

Pupil Interface

Strongly disagree				Strongly agree
	√			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
√				
1	2	3	4	

---

7. I would imagine that most people would learn to use this system very quickly

Pupil Interface

Strongly disagree				Strongly agree
			√	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
				√
1	2	3	4	

---

8. I found the system very cumbersome to use

Pupil Interface

Strongly disagree				Strongly agree
	√			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
√				
1	2	3	4	

---

9. I felt very confident using the system

Pupil Interface

Strongly disagree				Strongly agree
			√	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
			√	
1	2	3	4	

---

10. I needed to learn a lot of things before I could get going with this system

Pupil Interface

Strongly disagree				Strongly agree
√				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
√				
1	2	3	4	

Please comment on aspects of the **teacher** interface that didn't work properly or could be improved.

Please comment on aspects of the **pupil** interface that didn't work properly or could be improved.

Fine once I was used to positioning of instructions, info

Fitness for teaching and learning

If there were more worked examples in WEAVE would you use it in your classes regularly? Please explain your answer.

Yes, I would like to see some really basic and small Python programs as a starter going through basic constructs on their own so could start S3/N4 level and introduce interface at same time. The key to programming is trying to get pupils to slow down, think very simply, give real time to analysis, I like how the tasks deliberately show the step-by-step process required to first analyse a problem then program it. I think the time on questions is very interesting to look at as some should have longer times – pupils could skip through parts, maybe some sections should have set time where pupil can't skip on screen to try and encourage them to really focus on program/instructions.

Presentation of pupils' usage data

How easy was it to understand what information did the different graphs/tables show?

Very difficult Very easy

			√	
1	2	3	4	5

How useful did you find the graphs/tables in understanding how your pupils worked with the examples?

Not useful at all Very useful

			√	
1	2	3	4	5

Is there any other information that the system does not provide but would help you to better understand the difficulties of your pupils?

Button to click (traffic light) at set sections so pupils can say what know/kind of know/ don't know – teacher can see mood of class on different parts and pupils can show need help more privately

Deploying WEAVE in schools

How easy do you think would be to deploy WEAVE in your school?

Very difficult Very easy

				√
1	2	3	4	5

Logistics

Are there any logistical problems with using WEAVE in your classroom? Possible issues are: website blocking, browser incompatibilities, machine limitations (speed, screen size etc.)

No seems to work fine

Other comments

Do you have any other comments to make about the system? For example, most of these questions ask about problems… in addition, what do you like about the system?!

I really like it – encouraging and showing step by step working through analysis in particular we skirt over too much, pupils are getting more used to working online with systems like Code.org and CodeAcademy and they really like it, teacher progress check helps encourage them to work – some continue at home, like the time spent for us to see who's really engaging You could ask longer answers typed in, I like how you could select a specific question and see all the whole class responses to see how they explain it together, pupils like to see how they all describe programs differently Assumptions part of analysis is good, always need to think between the lines of a problem to work out what else would be needed in the program I'm interested in how we could assess analysis/development skills in Computing better – a system like this? I think the debate currently raging about internal/external marking of projects misses the point that a 15 page report with a working database/web site/program that is just an end product is not how to assess computational thinking skills of a pupil. We need to be able to assess the pupil's "journey" to the final product.

E.2.2.2 Participant 2

Your use of WEAVE

Please tick which of the following uses you have made of WEAVE (as many as apply)

1. x I viewed worked examples myself using the pupil interface
2. ____ I looked at the teacher interface
3. x I used WEAVE in one or more classes with pupils
4. ____ I used WEAVE with pupils and then used the teacher interface to explore their progress

Usability

The following is the standard SUS usability survey. When answering these, note that each question has space for two answers, one for the teacher interface and one for the pupil interface. If you didn't use one or other interface, just leave that part unanswered.

1. I think that I would like to use this system frequently

Pupil Interface

Strongly disagree				Strongly agree
			x	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	5

---

2. I found the system unnecessarily complex

Pupil Interface

Strongly disagree				Strongly agree
	x			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

3. I thought the system was easy to use

Pupil Interface

Strongly disagree				Strongly agree
			x	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

4. I think that I would need the support of a technical person to be able to use this system

Pupil Interface

Strongly disagree				Strongly agree
x				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

5. I found the various functions in this system were well integrated

Pupil Interface

Strongly disagree				Strongly agree
			x	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

6. I thought there was too much inconsistency in this system

Pupil Interface

Strongly disagree				Strongly agree
	x			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

7. I would imagine that most people would learn to use this system very quickly

Pupil Interface

Strongly disagree				Strongly agree
				x
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

8. I found the system very cumbersome to use

Pupil Interface

Strongly disagree				Strongly agree
x				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

9. I felt very confident using the system

Pupil Interface

Strongly disagree				Strongly agree
				x
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

10. I needed to learn a lot of things before I could get going with this system

Pupil Interface

Strongly disagree				Strongly agree
x				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

Please comment on aspects of the **teacher** interface that didn't work properly or could be improved.

Please comment on aspects of the **pupil** interface that didn't work properly or could be improved.

The worked examples are very verbose and pupils here at Kirkland had and will continue to have difficulty with the comprehension of the text/feedback/questions etc.

Fitness for teaching and learning

If there were more worked examples in WEAVE would you use it in your classes regularly? Please explain your answer.

I would welcome a development of this nature however the issues noted in my very brief experience include: Python worked examples are Python 2 and Python 3 is used in class (doesn't add confusion but just gives another thought process for comprehension pupils don't need at this stage of revision) Pupils were looking to have the code in a format they code run to help exemplify and understand the code. Examples made reference to modules not currently installed so prevented the running of the code to help exemplify and visualise the program. The wording and structure of the questions and answers would need to be simplifies to aid comprehension of the question before trying to understand the programming code.

Presentation of pupils' usage data

How easy was it to understand what information did the different graphs/tables show?

Very difficult				Very easy
1	2	3	4	5

How useful did you find the graphs/tables in understanding how your pupils worked with the examples?

Not useful at all				Very useful
1	2	3	4	5

Is there any other information that the system does not provide but would help you to better understand the difficulties of your pupils?

Sorry Quintin I didn't get to this stage.

Deploying WEAVE in schools

How easy do you think would be to deploy WEAVE in your school?

Very difficult				Very easy
				x
1	2	3	4	5

Web based, as it is, would not present any issues. However many councils operate policies which will restrict the storage of pupil information on a computer system which is not under the ownership of the council. Many

teachers frequently flaunt with this issue and are unaware of having pupil data stored which technically breaks this policy.

Logistics

Are there any logistical problems with using WEAVE in your classroom? Possible issues are: website blocking, browser incompatibilities, machine limitations (speed, screen size etc.)

It appeared to work straight from the tin, no issues but I've only used it once so far.

Other comments

Do you have any other comments to make about the system? For example, most of these questions ask about problems… in addition, what do you like about the system?!

I like this Quitin and would welcome a development in this area. As I've stated above I'd like to see the verbose nature of the content reduced, and simplified for all candidates/students. It's a kin to the Scholar materials which are to academic for pupils here at Kirkland. It would be nice to have code which either runs in the browser to demonstrate the program or provide the specifications for language the code would run under should pupils want to copy and paste it into their own environment. One issue is the languages used across schools, this will vary and therefore not meet the needs of most, especially if it's in python. I'm happy though!!

E.2.2.3 Participant 3

Your use of WEAVE

Please tick which of the following uses you have made of WEAVE (as many as apply)

1. _X___ I viewed worked examples myself using the pupil interface
2. ____ I looked at the teacher interface
3. ____ I used WEAVE in one or more classes with pupils
4. ____ I used WEAVE with pupils and then used the teacher interface to explore their progress

Usability

The following is the standard SUS usability survey. When answering these, note that each question has space for two answers, one for the teacher interface and one for the pupil interface. If you didn't use one or other interface, just leave that part unanswered.

1. I think that I would like to use this system frequently

Pupil Interface

Strongly disagree				Strongly agree
				x
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	5

---

2. I found the system unnecessarily complex

Pupil Interface

Strongly disagree				Strongly agree
x				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree

1	2	3	4

3. I thought the system was easy to use

Pupil Interface

Strongly disagree				Strongly agree
			x	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

4. I think that I would need the support of a technical person to be able to use this system

Pupil Interface

Strongly disagree				Strongly agree
x				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

5. I found the various functions in this system were well integrated

Pupil Interface

Strongly disagree				Strongly agree
				x
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

6. I thought there was too much inconsistency in this system

Pupil Interface

Strongly disagree				Strongly agree
x				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

7. I would imagine that most people would learn to use this system very quickly

Pupil Interface

Strongly disagree				Strongly agree
			x	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

8. I found the system very cumbersome to use

Pupil Interface

Strongly disagree				Strongly agree
x				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

9. I felt very confident using the system

Pupil Interface

Strongly disagree				Strongly agree
				x
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

10. I needed to learn a lot of things before I could get going with this system

Pupil Interface

Strongly disagree				Strongly agree
x				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

Please comment on aspects of the **teacher** interface that didn't work properly or could be improved.

Please comment on aspects of the **pupil** interface that didn't work properly or could be improved.

The match between the pupil answer and the system answer could perhaps be made clearer. It would be useful to label the three parts of the interface (eg problem description/specification;

coding and Help/ Step by Step)

Fitness for teaching and learning

If there were more worked examples in WEAVE would you use it in your classes regularly? Please explain your answer.

Yes. At the moment I am teaching National 5 Computing Science. I think that I would use some simpler examples with my classes. The examples currently in the system would be useful for Higher.

Presentation of pupils' usage data

How easy was it to understand what information did the different graphs/tables show?

Very difficult				Very easy
1	2	3	4	5

How useful did you find the graphs/tables in understanding how your pupils worked with the examples?

Not useful at all				Very useful
1	2	3	4	5

Is there any other information that the system does not provide but would help you to better understand the difficulties of your pupils?

Deploying WEAVE in schools

How easy do you think would be to deploy WEAVE in your school?

Very difficult				Very easy
				x
1	2	3	4	5

Logistics

Are there any logistical problems with using WEAVE in your classroom? Possible issues are: website blocking, browser incompatibilities, machine limitations (speed, screen size etc.)

I would need to try it out in the classroom using a pupil id. Even if the filtering system blocked it, I can ask for the url to be allowed, then I think it would be OK to access. The browser we use in school is Internet Explorer.

Other comments

Do you have any other comments to make about the system? For example, most of these questions ask about problems… in addition, what do you like about the system?!

I've tried it, from the pupil side, on my ipad mini and on a PC using Chrome and it worked very well. The step by step approach is very good,, and the addition of the multiple choice questions will make pupils think actively, rather than just reading through a worked example. Although there are a lot of steps, this will help to identify exactly where a problem in understanding is. The problems would be too difficult at the moment for my pupils who are National 5, but would be excellent for Higher. I like the way that your system highlights the problem description on the Left hand side, as it is explained in the lower panel.
Look forward to exploring it further (and more National 5 problems please!).

Another point – it would be useful to have a very basic example of the algorithm as well as where it is embedded in a problem .eg simply a linear search as well as the example with the door entry.

E.2.2.4 Participant 4

Your use of WEAVE

Please tick which of the following uses you have made of WEAVE (as many as apply)

1. I viewed worked examples myself using the pupil interface √
2. I looked at the teacher interface √
3. I used WEAVE in one or more classes with pupils
4. I used WEAVE with pupils and then used the teacher interface to explore their progress

Usability

The following is the standard SUS usability survey. When answering these, note that each question has space for two answers, one for the teacher interface and one for the pupil interface. If you didn't use one or other interface, just leave that part unanswered.

1. I think that I would like to use this system frequently

Pupil Interface

Strongly disagree				Strongly agree
		√		
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	5

---

2. I found the system unnecessarily complex

Pupil Interface

Strongly disagree				Strongly agree
	√			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

3. I thought the system was easy to use

Pupil Interface

Strongly disagree				Strongly agree
			√	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

4. I think that I would need the support of a technical person to be able to use this system

Pupil Interface

Strongly disagree				Strongly agree
√				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

5. I found the various functions in this system were well integrated

Pupil Interface

Strongly disagree				Strongly agree
			√	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

6. I thought there was too much inconsistency in this system

Pupil Interface

Strongly disagree				Strongly agree
	√			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

7. I would imagine that most people would learn to use this system very quickly

Pupil Interface

Strongly disagree				Strongly agree
				√
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

8. I found the system very cumbersome to use

Pupil Interface

Strongly disagree				Strongly agree
	√			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

9. I felt very confident using the system

Pupil Interface

Strongly disagree				Strongly agree
			√	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

---

10. I needed to learn a lot of things before I could get going with this system

Pupil Interface

Strongly disagree				Strongly agree
√				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
1	2	3	4	

Please comment on aspects of the **teacher** interface that didn't work properly or could be improved.

Please comment on aspects of the **pupil** interface that didn't work properly or could be improved.

Pop up question windows can be closed with no feedback. There does not seem to be way for a user to navigate back and have the pop up window open again.

The three presentation windows seem to provide a clear view of what is going on but probably should be titled to give users some idea of what to expect. For example, initially at least, the right hand window does nothing and it is not clear whether it might do something later.

When a pop up option is selected and then submitted you don't seem to get any feedback as to whether an answer submitted is correct or not and this feedback is pretty essential in supporting learning. Pupils may ask for instance, 'is my function name OK as it does match the example shown', or 'did I choose the correct option in a list of options', etc. In the case of free form responses, feedback could be given by altering the question style to a multiple choice with explanation as to why options from which to choose are or are not appropriate, eg naming a function.

When explaining code or the thinking behind how the code was generated, it might be better to reference the 'variable names' in the code– see slide 21 where 'ids array' is written instead of 'authUsers' .

Fitness for teaching and learning

If there were more worked examples in WEAVE would you use it in your classes regularly? Please explain your answer.

I would use an adapted form of this system with classes in a variety of ways to support my own style in teaching programming. It is a resource that has the potential to be used flexibly and with a range of pupil ability, experience and knowledge.

Presentation of pupils' usage data

How easy was it to understand what information did the different graphs/tables show?

Very difficult				Very easy
1	2	3	4	5

How useful did you find the graphs/tables in understanding how your pupils worked with the examples?

Not useful at all				Very useful
1	2	3	4	5

Is there any other information that the system does not provide but would help you to better understand the difficulties of your pupils?

We seem to have replaced the 'Sage On Stage' with an approach of the 'Sage In The New Age '. Why have you not gone instead for replacing the 'Guide On The Side' with the 'Guide On The Wide' <area network> - epsilon. Build some degree of intelligence that can interpret user responses and tailor feedback to learner need; starting with at the very least feedback on whether a user response is correct or not and moving towards the more challenging adaptive response style interface. As a teacher, what I don't want to have to be doing with classes is explaining the explanations.

Deploying WEAVE in schools

How easy do you think would be to deploy WEAVE in your school?

Very difficult				Very easy
				√
1	2	3	4	5

Logistics

Are there any logistical problems with using WEAVE in your classroom? Possible issues are: website blocking, browser incompatibilities, machine limitations (speed, screen size etc.)

Not that I can see

Other comments

Do you have any other comments to make about the system? For example, most of these questions ask about problems… in addition, what do you like about the system?!

The questions about usability seem to be very summative and quite general, ie top down. What is really needed at this stage is perhaps more formative questions to collect detailed information that might aid in the iteration of the prototype. It might therefore be useful to add, eg a 'feedback button' to the tutorial pages at least for a little while, to allow users the opportunity to describe their experience and to comment on particular slides, both technical and usability comments. This would provide useful formative evaluation data.

E.2.2.5 Participant 5

Your use of WEAVE

Please tick which of the following uses you have made of WEAVE (as many as apply)

1. _✓_ I viewed worked examples myself using the pupil interface
2. _✓__ I looked at the teacher interface
3. _✓__ I used WEAVE in one or more classes with pupils
4. ____ I used WEAVE with pupils and then used the teacher interface to explore their progress

Usability

The following is the standard SUS usability survey. When answering these, note that each question has space for two answers, one for the teacher interface and one for the pupil interface. If you didn't use one or other interface, just leave that part unanswered.

1. I think that I would like to use this system frequently

Pupil Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	5

---

2. I found the system unnecessarily complex

Pupil Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

3. I thought the system was easy to use

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	

---

4. I think that I would need the support of a technical person to be able to use this system

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	

---

5. I found the various functions in this system were well integrated

Pupil Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

6. I thought there was too much inconsistency in this system

Pupil Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

7. I would imagine that most people would learn to use this system very quickly

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

8. I found the system very cumbersome to use

Pupil Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

9. I felt very confident using the system

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

10. I needed to learn a lot of things before I could get going with this system

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	

Please comment on aspects of the **teacher** interface that didn't work properly or could be improved.

Just more intuitive; Simpler to use, more instructions on how to locate things Maybe link in with local authority systems to get class lists put in for you Reporting feature Link with the TRACE task (PLAN C) to make that electronic.

Please comment on aspects of the **pupil** interface that didn't work properly or could be improved.

FROM pupils themselves;

Cons

- Old fashioned aesthetics
- Doesn't allow us to engage very well
- Questions are not 100% clear
- Window questions does not move smoothly – at times it sticks
- "numbers Generated" variable not used in code in one of the exercises
- One of the exercises jumped from step 3 to step 6
- Would be good to explain some of the computational terms – selection statements, one arm if statement, two arm if statements, multiple option selections – not language pupils would use
- Could be more personal to help the user to feel included in the system to help engagement e.g. Now you should try…
- Could be more colourful.

Pros

- Highlights areas of focus with the user
- Instructions in related problems are clear.

Feedback from a few pupils – sorry if it is quite critical. I have just directly fed some of the feedback back to you.

Fitness for teaching and learning

If there were more worked examples in WEAVE would you use it in your classes regularly? Please explain your answer.

Yes I would. I feel working through examples will only consolidate the pupils learning.

Presentation of pupils' usage data

How easy was it to understand what information did the different graphs/tables show?

Very difficult				Very easy
1	2	3	4	5

How useful did you find the graphs/tables in understanding how your pupils worked with the examples?

Not useful at all				Very useful
1	2	3	4	5

Is there any other information that the system does not provide but would help you to better understand the difficulties of your pupils?

Deploying WEAVE in schools

How easy do you think would be to deploy WEAVE in your school?

Very difficult				Very easy
			✓	
1	2	3	4	5

Logistics

Are there any logistical problems with using WEAVE in your classroom? Possible issues are: website blocking, browser incompatibilities, machine limitations (speed, screen size etc.)

Accessible in schools, no website blocking or system problems.

Other comments

Do you have any other comments to make about the system? For example, most of these questions ask about problems… in addition, what do you like about the system?!

None at minute – would be prepared to talk more about this at later stage.

E.3 Heuristics Evaluation

This Appendix contains the raw feedback from the heuristics evaluation that was conducted as part of this project.

Heuristics Evaluation Questions

1. Visibility of system status

- Did you receive enough feedback of the system's status?
  Yes – both implicit and explicit feedback was sufficient in understanding the system's state. An example of explicit feedback – the confirmation message in green when my teacher account had been successfully registered. Implicit feedback included the greying out of the "next" button in the student interface, when the end of the exercise had been reached.

2. Match between system and real world

- Is the terminology used familiar to you?
  Yes, the terminology was not an issue whatsoever.
- Does the interface present information in a simple, natural and logical order?
  I think so, things make sense, and the page I saw after clicking a link was pretty much what I expected to see.
- Can you easily understand the use of metaphors?
  Do you mean like the icons as a metaphor? If that is what you mean, yes, they make perfect sense.

3. User control and freedom

- Do you feel in control with the system?
  Yes. Things don't go mental.
- Is there anything that you think restricted you to complete a task?
  Not particularly, I understood how to interact with the system fine.
  Do you think that the facilities provided to return to the top level at any stage are appropriate?
  Yes. You click the homepage icon. Easy.
- Do you think that the facilities provided to recover from an error were appropriate?
  Well, if an error occurred, it was only due to me entering wrong input. So yes, the input boxes were still provided to me so I could correct the error.

4. Consistency and standards

- Do you think that the terminology and the controls are consistent throughout the system?
  Generally, yes, controls look consistent across different pages of the app.

- Is the look and feel of the system consistent?
  I think there's a few little things which could be improved in terms of UI consistency. For example, little things like the "home" icon being different sizes for the teacher interface and student interface. In the teacher interface, consistency in the statistics page would be good for the drop down boxes and the application selection dropbox would be good. Little things like that, if fixed, would improve the experience of the application dramatically.

5. Error prevention

- Are the input options chosen suitably to prevent errors?
  Yes – dropdowns for a finite set of choices. Input text boxes for usernames/passwords etc.
- Recognition rather than recall
  ?? I recognise that invalid input has been supplied…with the red text that appears. I don't fully understand this point.
- *Were you able to access help and instructions easily when needed?*
  I didn't see much help/instructions. I could see the little step-by-step tutorial in the student interface, but that was it. In that case, no.
- *Did you* find the relationship between controls and their actions obvious?
  Yes. The icons used for the controls were self-explanatory.

6. Flexibility and efficiency of use

- Does the website provide any shortcuts for improving interactions?
  Yup! You can use the left/right arrow keys in the student interface to go back/advance through steps – this is intuitive and a nice feature.
- Does the website interface guide you how to use it sufficiently?
  Yes, makes sense. And the step-by-step guide is sufficient, too (at least for the student interface). For the teacher interface, the main page seems to make sense on what you can do/what you need to do.
- Did you feel some of the registrations were unnecessary?
  What do you mean by this? If you can use the system without registering, then maybe they are. But I understand you need to register/sign up to record interactions.

7. Aesthetic and minimalist design

- Did you find the design simple, easy to learn and intuitive?
  It was fine – like many other websites out there today, with a navbar at the top and the main interface below.
- Do the interfaces have any information which you found irrelevant and distracting?
  Not to my knowledge. I felt the system was simplistic and there was no obvious clutter.
- Are the different controls of the system clearly labelled and obvious?
  Yeah, I think so – it's mainly the pictorial representations (icons) that make it easy to see what different things do. And for me, these were fine.

8. Help users recover from errors

- Did you find any error messages useful to help you recover from the error?
  Yes – trying out the statistics page, I deliberately didn't select something from one of the drop downs. The red text "Invalid Selection" is fine to highlight this. Also, same goes for logging in.
- Was it clear by the error messages what you have done wrong?
  Yes. Simple, to the point.

9. Help and documentation

- Is sufficient help provided?
  I only saw the little step-by-step guide for the student interface, ironically the easier of the two. I think some more assistance would be beneficial for some users of the teacher interface.

- Are the help messages clear enough for you to understand them?
  From what I saw, yes.

10. Navigation

- Did you feel at any point that navigation in the website was inappropriate or unnecessary?
  No, it all seemed reasonable to me.
- Was it clear to you what page you are on and where can you go next?
  I think so, but the only thing I was unsure about was the questions in the student interface. Can you actually not answer if you don't want to? What happens?

11. Structure of information

- Did you feel that related pieces of information are clustered together?
  Yes. Navigation links are together, for example.
- Is the size of the different elements appropriate in respect to the size of the screen and their importance?
  Yes – text is well-sized. Icons are well-sized, and match the size of the associated text. The interface fills the browser viewport, so you are making maximum use of the space available.

12. Physical constraints

- Is the distance between different elements and their size appropriate?
  I think so, it does not take too much effort to navigate to the next page on an exercise, for example.

E.4 Practical Evaluation

This Appendix contains the results from the practical evaluation conducted with Mr. Peter Donaldson and Mr Craig Henderson.

E.4.1 Peter Donaldson

Your use of WEAVE

Please tick which of the following uses you have made of WEAVE (as many as apply)

1. _✓_ I viewed worked examples myself using the pupil interface
2. _✓_ I looked at the teacher interface
3. _✓_ I used WEAVE in one or more classes with pupils
4. _✓__ I used WEAVE with pupils and then used the teacher interface to explore their progress

Usability

The following is the standard SUS usability survey. When answering these, note that each question has space for two answers, one for the teacher interface and one for the pupil interface. If you didn't use one or other interface, just leave that part unanswered.

1. I think that I would like to use this system frequently

Pupil Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

The main option screen for the teacher interface is ok however the way that the statistics section is organised makes it difficult to analyse the data and explore trends and patterns.

2. I found the system unnecessarily complex

Pupil Interface

Strongly disagree				Strongly agree
✓				
1	2	3	4	5

Teacher Interface

Strongly disagree Strongly agree

			✓	
1	2	3	4	

The way in which the statistics section was organised made it more difficult to analyse. I would have liked the option to export all of the class data into excel so I could do some more detailed analysis. The pupil id's was ok but when I copied and pasted the table into excel it didn't work unless I selected paste data only.

3. I thought the system was easy to use

Pupil Interface

Strongly disagree Strongly agree

			✓	
1	2	3	4	5

Teacher Interface

Strongly disagree Strongly agree

		✓		
1	2	3	4	

There was no easy way to move from the teacher interface back to the pupil interface without url hacking. When doing a demonstration being able to move between the two would have been useful.

4. I think that I would need the support of a technical person to be able to use this system

Pupil Interface

Strongly disagree Strongly agree

✓				
1	2	3	4	5

Teacher Interface

Strongly disagree Strongly agree

		✓		
1	2	3	4	

Appears as if you can add additional students but you have to delete the whole group which could cause problems if someone accidentally adds too many people to a group.

5. I found the various functions in this system were well integrated

Pupil Interface

Strongly disagree Strongly agree

			✓	
1	2	3	4	5

Teacher Interface

Strongly disagree Strongly agree

✓				
1	2	3	4	

The teacher interface needed a better level of task analysis as a number of potential difficulties could arise in certain situations. I can register a new group but it also appears as if I can register a group with the same name- there is no visual feedback to say that group already exists. Will it stop me from doing that or will it overwrite the existing group's data?

I can update the number of students in an existing group but I can't delete particular users from a group. Also I can view the group list but I have to click back on the browser in order to be able to change to a different group.

6. I thought there was too much inconsistency in this system

Pupil Interface

Strongly disagree Strongly agree

	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree Strongly agree

			✓	
1	2	3	4	

---

7. I would imagine that most people would learn to use this system very quickly

Pupil Interface

Strongly disagree Strongly agree

				✓
1	2	3	4	5

Teacher Interface

Strongly disagree Strongly agree

			✓	
1	2	3	4	

There are a limited number of options so it isn't difficult to learn how to use it.

8. I found the system very cumbersome to use

Pupil Interface

Strongly disagree				Strongly agree
✓				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	

---

9. I felt very confident using the system

Pupil Interface

Strongly disagree				Strongly agree
				✓
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	

---

10. I needed to learn a lot of things before I could get going with this system

Pupil Interface

Strongly disagree				Strongly agree
✓				
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	

Please comment on aspects of the **teacher** interface that didn't work properly or could be improved.

Register/Update/Delete Section
Issues around updating or deleting an existing group which could cause problems. I should be able add or remove new ids to a particular group or add or remove a whole set.

View group
I would like to be able to open the list up in Excel so that I can quickly record names and save a local copy. Copying and pasting from the browser into Word is fine however Excel is a bit fiddly.

I should be able to enter first names and last names against a student id in WEAVE. These could be encrypted on the server using my password so that the personally identifiable data was only accessible by the teacher who created that set.

View Statistics
I should be able to export a nicely organised set of data for the whole class in excel or csv format.

Selecting an example
All examples should not be listed instead just the examples where there is data for this particular group should be listed.

When I select a new view it blanks the worked example and I need to reselect the example again. I should be able to change views without having to reselect the same example each time.

At the moment all of the lists are active but you should have to select a year first then the group should be active then worked example and finally the type of data you would like to view. This would mean each list is activated only after a selection has been made in the previous one. The order of the lists would have to be academic year, group, worked example then data view and not data view, worked example which is what it currently is.

General Graph Improvements
I'd like different coloured bars to highlight pupils who are well above the average response time or well below for a particular step as well as the colour of their id. I'd also like answer graphs to show the correct answer choice in green.

Class Steps
I'd like the explanation text for the step displayed at the top of the graph as well as the step number.

Class Summary
The data is useful however I would like the total time to be displayed in minutes and seconds if it's above 59 seconds. I'd also like to be able to click on a pupil id or their total time to get a break down of how long they took for each step. It would be useful to be able to sort this table in different orders

such as by

Average Time
Average time is ok although the time labels and number of steps overlap too much and possibly should just be displayed when you hover over a node for more information.

Please comment on aspects of the **pupil** interface that didn't work properly or could be improved.

Horizontal scroll bars appeared if the name of the worked example was longer than the width of the drop down list instead of it being resized.

Application should be renamed to Example and each example should list the number of languages it's available in instead of having the language as part of the name.

I should be able to use the pupil interface as a teacher without having to use a pupil id. Anything I do during a demonstration would not be recorded and would not appear against a particular class set.

For a particular group I would like to be able to pick particular worked examples and the order in which they should be listed for pupils in that group so that they only see those examples and not every available example.

Fitness for teaching and learning

If there were more worked examples in WEAVE would you use it in your classes regularly? Please explain your answer.

Yes I would use WEAVE with my Higher and Advanced Higher pupils however the placement on the step text may be better as audio or possibly should be above the example and not below the two other panes. Experimental data would need to be gathered to see which arrangement was better.

Presentation of pupils' usage data

How easy was it to understand what information did the different graphs/tables show?

Very difficult				Very easy
	✓			
1	2	3	4	5

How useful did you find the graphs/tables in understanding how your pupils worked with the examples?

Not useful at all				Very useful
	✓			
1	2	3	4	5

Is there any other information that the system does not provide but would help you to better understand the difficulties of your pupils?

The way individual graphs and data screens are organised and presented would need to be refined or in some cases redesigned to get maximum benefit from this section of weave.

Deploying WEAVE in schools

How easy do you think would be to deploy WEAVE in your school?

Very difficult				Very easy
				✓
1	2	3	4	5

Logistics

Are there any logistical problems with using WEAVE in your classroom? Possible issues are: website blocking, browser incompatibilities, machine limitations (speed, screen size etc.)

There are some resizing glitches where if a pupil zooms in to the page to make the text bigger the panes end up overlapping or leaving lines over the other pane making the text more difficult to see. On smaller screen resolutions WEAVE might be difficult to use.

Other comments

Do you have any other comments to make about the system? For example, most of these questions ask about problems… in addition, what do you like about the system?!

The potential of the system is really exciting and I hope that further work is done to develop it. Some of the features are really neat however several rounds of paper prototyping with different teachers need to be conducted in order to create a better experience in the teacher interface section. Individually there is nothing wrong with some of the data views but in practice making use of this data to improve learning and teaching would be difficult. This is something that many commercial systems don't necessarily get right either in version 1.

E.4.2 Craig Henderson

Your use of WEAVE

Please tick which of the following uses you have made of WEAVE (as many as apply)

1. _✓_ I viewed worked examples myself using the pupil interface
2. _✓__ I looked at the teacher interface
3. _✓__ I used WEAVE in one or more classes with pupils
4. ____ I used WEAVE with pupils and then used the teacher interface to explore their progress

Usability

The following is the standard SUS usability survey. When answering these, note that each question has space for two answers, one for the teacher interface and one for the pupil interface. If you didn't use one or other interface, just leave that part unanswered.

1. I think that I would like to use this system frequently

Pupil Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	5

---

2. I found the system unnecessarily complex

Pupil Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

3. I thought the system was easy to use

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	

---

4. I think that I would need the support of a technical person to be able to use this system

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	

---

5. I found the various functions in this system were well integrated

Pupil Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

6. I thought there was too much inconsistency in this system

Pupil Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

7. I would imagine that most people would learn to use this system very quickly

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

8. I found the system very cumbersome to use

Pupil Interface

Strongly disagree				Strongly agree
			✓	
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

9. I felt very confident using the system

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
		✓		
1	2	3	4	

---

10. I needed to learn a lot of things before I could get going with this system

Pupil Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	5

Teacher Interface

Strongly disagree				Strongly agree
	✓			
1	2	3	4	

Please comment on aspects of the **teacher** interface that didn't work properly or could be improved.

Just more intuitive; Simpler to use, more instructions on how to locate things Maybe link in with local authority systems to get class lists put in for you Reporting feature Link with the TRACE task (PLAN C) to make that electronic.

Please comment on aspects of the **pupil** interface that didn't work properly or could be improved.

FROM pupils themselves;

Cons

- Old fashioned aesthetics
- Doesn't allow us to engage very well
- Questions are not 100% clear
- Window questions does not move smoothly – at times it sticks
- "numbers Generated" variable not used in code in one of the exercises
- One of the exercises jumped from step 3 to step 6
- Would be good to explain some of the computational terms – selection statements, one arm if statement, two arm if statements, multiple option selections – not language pupils would use
- Could be more personal to help the user to feel included in the system to help engagement e.g. Now you should try…
- Could be more colourful.

Pros

- Highlights areas of focus with the user
- Instructions in related problems are clear.

Feedback from a few pupils – sorry if it is quite critical. I have just directly fed some of the feedback back to you.

Fitness for teaching and learning

If there were more worked examples in WEAVE would you use it in your classes regularly? Please explain your answer.

Yes I would. I feel working through examples will only consolidate the pupils learning.

Presentation of pupils' usage data

How easy was it to understand what information did the different graphs/tables show?

Very difficult				Very easy
1	2	3	4	5

How useful did you find the graphs/tables in understanding how your pupils worked with the examples?

Not useful at all				Very useful
1	2	3	4	5

Is there any other information that the system does not provide but would help you to better understand the difficulties of your pupils?

Deploying WEAVE in schools

How easy do you think would be to deploy WEAVE in your school?

Very difficult				Very easy
			✓	
1	2	3	4	5

Logistics

Are there any logistical problems with using WEAVE in your classroom? Possible issues are: website blocking, browser incompatibilities, machine limitations (speed, screen size etc.)

Accessible in schools, no website blocking or system problems.

Other comments

Do you have any other comments to make about the system? For example, most of these questions ask about problems… in addition, what do you like about the system?!

None at minute – would be prepared to talk more about this at later stage.

Appendix F

Email from Mr. Peter Donaldson

Hello Emilia,

Attached to this email is my completed WEAVE questionnaire in .docx format and .rtf in case you have difficulties with the .docx version.

I have written quite a bit that's focused around bugs and improvements but that's not because the existing implementation is bad it's just that there is so much potential for using this system with Computing classes.

The system is something I would expect a team of people to have produced over a much longer period so the fact you've managed to get so much functionality as a student is really impressive.

hope you find this useful,
Peter

Peter Donaldson
National Project Officer **PLAN C**

Bibliography

[1] Robert K. Atkinson, Sharon J. Derry, Alexander Renkl, and Donald Wortham. Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, 70:181–214, 2000.

[2] Aaron Bangor, Philip T. Kortum, and James T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24, 2008.

[3] R.C. Clark, F. Nguyen, and J Sweller. *Efficiency in Learning: Evidence-Based Guidelines to Manage Cognitive Load.* Pfeiffer, 2006.

[4] Ruth Colvin Clark and Richard E. Mayer. *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*. Pfeiffer & Company, 2nd edition, 2007.

[5] Allan Collins, John Seely Brown, and Susan E. Newman. Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics. Technical Report 403, Center for the Study of Reading, January 1987.

[6] R.L. Cooper, L. Zhao, and C. Wang. A model driven architecture and toolset for building immersive software engineering teaching tools. In *European Conference on Electronic Learning, Copenhagen*, 2007.

[7] Hans Jonasson. *Determining Project Requirements, Second Edition: Mastering the BABOK and the CBAP Exam*. Auerbach Publications, 2nd edition, 2012.

[8] Allan MacLean, Richard M. Young, Victoria Bellotti, and Thomas P. Moran. Questions, options, and criteria: Elements of design space analysis. *Human-Computer Interaction*, 6, 1991.

[9] Fred Paas, Alexander Renkl, and John Sweller. Cognitive Load Theory and Instructional Design: Recent Developments. *Educational Psychologist*, 38, 2003.

[10] Stephen K. Reed and Cheryl Bolstad. Use of examples and procedures in problem solving. *Journal of Experimental Psychology Learning Memory and Cognition*, 17, 1991.

[11] Alexander Renkl. The worked-out examples principle in multimedia learning. In Richard Mayer, editor, *The Cambridge Handbook of Multimedia Learning*, pages 229–246. Cambridge University Press, 2005.

[12] Alexander Renkl, Robert K. Atkinson, and Cornelia S. Große. How fading worked solution steps works - a cognitive load perspective. *Instructional Science*, 32, 2004.

[13] The Royal Society. Shut down or restart? the way forward for computing in uk schools, 2012.

[14] Yulun Song. *An Authoring and Presentation Environment for Interactive Worked Examples*. PhD thesis, University of Glasgow, August.

[15] John Sweller and Graham A. Cooper. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2(1):pp. 59–89, 1985.

[16] John Sweller, Jeroen J. G. Van Merrienboer, and Fred G. W. C. Paas. Cognitive architecture and instructional design. *Educational Psychology Review*, 10:251–296, 1998.

[17] Rohani A. Tarmizi and John Sweller. Guidance during mathematical problem solving. *Journal of Educational Psychology*, 80(4):pp. 424–436, 1988.

[18] J.J.G. van Merriënboer. *Training Complex Cognitive Skills: A Four-Component Instructional Design Model for Technical Training*. Educational Technology Publications, 1997.

[19] Robert A. Virzi. Refining the test phase of usability evaluation: How many subjects is enough? *Hum. Factors*, 34, 1992.

[20] Mark Ward and John Sweller. Structuring effective worked examples. *Cognition and Instruction*, 7(1):1–39, 1990.

[21] Xinming Zhu and Herbert A. Simon. Learning mathematics from examples and by doing. *Cognition and Instruction*, 4(3):137–166, 1987.