Teaching programming is inherently difficult. Literature on learning suggests that the most efficient way to learn something is by practice. The way schools in the UK incorporate this approach 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 often novices lack a good enough understanding of the domain. This leads to their striving to get the right answer rather than gaining a better understanding of the domain.

A good way of teaching somebody a completely new way of thinking is by showing them the process of thought involved in solving a particular problem. 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 and time wouldn’t allow them to show their students what cognitive steps they should undertake in order to solve the posed problem.

Research has shown that step-by-step guidance of the process of solving problems can help novices gain better understanding of the solving techniques. There are books providing such guidance in the form of worked examples that have proven to be effective. 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), to address the posed issues mentioned above as well as to evaluate to what extent such an application will prove effective in lowering the learning curve for pupils in computing science. It consists of 2 interfaces- one for students and one for creators of worked examples. The student interface provides users with a selection of examples to choose from. These examples consist of a problem specification and a step-by-step guidance together with explanation of the process of thinking involved at each step. Questions may be asked to engage the user further as well as provide feedback to the creator of these examples of whether their approach at specifying the process of thinking enhances the understanding of the concept of interest or confuses the user. The other interface enables the creation of examples to accommodate a teacher’s specific needs.

Since the idea of IWE was to explore the extent to which it can fit in the teaching process in schools and whether it would be a potentially successful learning technique, the prototype wasn’t aiming at large scale deployment. The application has proven to be effective at enhancing the teaching of computing science in schools so the next logical step is to put it in use in the environment it is aimed for. 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 the IWE will pose to the whole system and submit a further request to a local authority responsible for the particular school. This gives the motivation of recreating IWE as a web-based application in order to start effectively presenting worked examples in larger context. This would avoid the complicated and time-consuming process of installing IWE in schools. Furthermore, they would be able to receive the latest updates of the application with no effort. The web-based version of IWE is called Worked Examples Viewer (WEAVE).

In addition to being the more easily deployable version of IWE, WEAVE takes a step further in improving the teaching process in schools by dividing its users into 3 groups. To the 2 user groups of IWE- the student and the author of worked examples- is added another group of users- the teachers. WEAVE provides a special teacher interface which shows information about the way students use the application. Teachers can examine in detail how pupils in their classes work with different examples. Information includes details about the time spent by students at each step of an example as well as answers to questions posed. Such information is aggregated into graphs which can be easily customised to the specific type of information the teacher wants to examine. This knowledge can help teachers to adjust their teaching material as well as make them aware of what pupils find most difficult and problematic so that they address this appropriately and on time. This is a feature that a book with worked examples can’t provide readers or the author of the book with. Spending more time at a step would be a sign to the teacher that this step needs to be explained further or to the author of the example that the explanation is unclear and needs revisiting. Evaluation will show how useful this would be for teachers to understand areas of difficulties and adjust their teaching practices accordingly.

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 to students to undertake further learning. 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 students and enable them to help each other to become better in teaching computing science.

The rest of this dissertation describes 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 the WEAVE’s successful integration in every day teaching practices in schools. The final chapter is dedicated to the future developments for the system which will be addressed shortly.