Best Practices in using Technological Infrastructures

Other

Best Practices in using Technological Infrastructures

Institute of Coding Deliverable D1.2.6

Oli Howson

July 2020

Contents

1 Introduction

This report follows on from the D1.2.5 report on the IoC Common Curricula and aims to provide an overview of how an agreed Institute of Code infrastructure has been, or is being, deployed for student benefit to enhance their industrial readiness.

The report consists of a number of case studies of investigations and activities which have been carried out or are underway within the Open University[1], primarily within the school of Computing and Communications. As such the focus is within the Higher Education sector, with a number of the case studies hinging upon the Jupyter infrastructure covered in the preceding report.

Due to the continually changing nature of the infrastructures available, and the developing nature of the work underway, this document is a snapshot at time of writing and it is expected that the document will need to be updated over time to maintain relevance.

In the D1.2.5 report we described the gap in digital skills. The Institute of Coding was set up by the UK government, through the office for students, to tackle this gap. Part of this challenge is in increasing the digital capabilities of learners by developing hardware infrastructure, software infrastructure, and pedagogical infrastructure, which will prepare them for industry and support other work streams within the IoC such as the accreditation of competency in recognising this readiness.

The D1.2.5 report was primarily an investigation into the affordances offered by a Jupyter ecosystem centric infrastructure and the hardware infrastructure needed to support that. Many of these lessons are being applied to the work done in TM351 (case study 2.2). However, many other projects within the Open University have also, often with IoC support, been investigating opportunities to develop or stretch physical, software or pedagogical infrastructure to promote the efficacy of learning, the diversity of audience and to increase the industry-readiness of the learner. Some of these projects are detailed in the following cases studies.

2 Case Studies

2.1 Pair Programming for Students Learning Programming at a Distance

Pair programming is a key aspect of Extreme Programming (XP), which encourages informal and immediate communication over joint coding work. The technique involves two developers: one in a driver role writes the code, controlling the keyboard and mouse, and the other, the navigator, reviews the code as the driver writes it. There is now a substantial body of evidence in support of improved coding outcomes when pair programming is used in teaching(Ghorashi & Jensen 2016, Celepkolu & Boyer 2018).

[1] https://www.open.ac.uk/

Despite these proven academic benefits for collocated classroom settings, distance and online settings are yet to benefit. The isolation and impoverished learning experience some students report points to an urgent need for tools and methods that support remote pair programming in a distributed educational setting to be developed.

In a traditional instructor-led teaching environment, pair programming is relatively easy to implement. This is more challenging in distance teaching and learning situations. Some of these challenges include:

1. Students experiencing isolation and exclusion
2. Problematic communication due to students having fewer interactive and communication options in a remote setting
3. Access to code repository

To improve learning programming at a distance, Pair-Programming as collaborative learning tool is being investigated and some initial literature research into this area has been investigated building on the known benefits of collaborative learning. The research is still on-going to include further findings about collaborative learning, methods and tools developed to support pair programming (both traditional and remote style) and real time coding in education (both collocated and distance settings) to identify principles of effective pair. There is currently an on-going user study that includes a website to coordinate the Pair Programming user study[2].

The user study has identified some challenges, including:

- Fewer than expected students signed up for the user studies.
- Out of these, a couple more students dropped out due to personal circumstances and COVID-19 related issues.

Preliminary analysis of student's journals showed a positive response to the Pair Programming approach. For example, a student revisited a programming task she had earlier completed alone with her partner. The student reported that the resulting output was better when she paired.

The current software tool used for the study is USE Together. This was specifically developed with professional programmers in mind. The study is further analysing the tools' usability for educational purposes. This will serve as a base for further studies on tools and infrastructure required for a successful implementation of Pair Programming at a Distance.

2.2 Jupyter infrastructure

In order to provide a complete environment for students to study data science, the module TM351[3] has shipped a USB stick containing a virtual machine for

[2]https://learn1.open.ac.uk/course/view.php?name=OURPP
[3]http://www.open.ac.uk/courses/modules/tm351

the 5 years of its presentation. The virtual machine approach has ensured that students were provided with a consistent programming environment, with minimal difficulties installing all the necessary software. This included much of the python data science and scientific software stack, as well as instances of the relational database PostgreSQL[4] and the document database MongoDB[5]. However, this has occasionally caused difficulties for students with unusual or outdated hardware configurations, requiring significant module team time to be spent supporting students on software installation.

The project, therefore, has been to develop an environment which can be hosted at the Open University, and which the students can immediately access without the need for a locally hosted VM (virtual machine). This also allows for students to access the materials with a lightweight computing platform such as a tablet or Chromebook, as it is no longer necessary to be able to execute code locally.

All programming in TM351 is handled via Jupyter[6] notebooks, and so the remote environment hosted at the Open University is almost identical to that shipped to students for local hosting.

The hardware hosting the remote environment was purchased with IoC funding and is managed internally by the STEM Specialist Support Unit (SSU). This runs a single instance of Jupyterhub[7], which allows each student to run any number of instances of notebooks. An instance of PostgreSQL and MongoDB are also hosted, on which students have individual accounts, also accessed via the notebooks running on Jupyterhub. This has allowed a service to be created which students can access directly using their University login details.

From a module perspective, the key challenge has been working, with SSU, to deliver a working framework. The need was not simply to deliver a working environment, but also to ensure that the environment provided was compatible with the various teaching materials which had been developed for the previous VM-based generations of the module. This is still an ongoing task: for the first cut of the hosted system, the environment was rigorously tested with the different resources from the module to determine challenges and inconsistencies and usability concerns. An iterative development and testing has taken place, using the teaching materials to ensure that they are fully compatible between versions. Appropriate modifications and iterative testing has been done to ensure a comparable teaching and learning experience in both environments.

Currently, both the remote version and the desktop version are being used by academic staff, tutors and students. The initial reports from Tutors when using the online version have been positive and over 50 percent of students are working with the online version of the environment (over a 100 students). A first module assessment submission has been achieved demonstrating the robust environment for students and tutors.

The infrastructure development was motivated by the intent is to increase

[4]https://www.postgresql.org/
[5]https://www.mongodb.com/
[6]https://jupyter.org/
[7]https://jupyter.org/hub

the quality of the educational experience of students by providing them with a professional and fully-functional learning environment to work effectively on data analysis and related techniques. This reduces the demand on students requiring to invest in high performance hardware. This may well help to reduce the barrier to access.

The infrastructure should potentially be usable for future IoC microcredentials or similar, just because we can install all the necessary resources (eg. a notebook, some sort of registration ticket, etc) at source and handle it there. We have coderunner[8] installed, so assessing the students' work automatically and issuing the necessary blockchain credential at our end should be feasibly possible.

2.3 Teaching distributed computing using Raspberry Pi clusters at a distance

TM129 Technologies in Practice is a compulsory Level 1 BSc (Honours) Computing & IT module at the Open University. It hosts around 1,800 students annually and is based on three topics – Robotics, Networking and Operating Systems.

This study provided the opportunity to give greater exposure to novel computing concepts, specifically computer clusters which provide large amounts of processing power to solve a range of everyday problems including decrypting data and image processing. One relatively-low cost approach to clusters is using everyday hardware such as the inexpensive Raspberry Pi computer. Our project focuses on exploring the benefits and challenges of teaching about computer clusters at a distance using low-cost Raspberry Pi clusters.

A review of the physical, software, or pedagogical infrastructure developed within or for your project.

Eleven custom-made Raspberry Pi clusters have been constructed by the module team, based on the OctaPi instructions[9] (apache 2 license). These will be hosted in the OpenSTEM Labs[10][11] and will be accessible by remote students over the Internet to perform a range of activities.

In the first instance, TM129 students will have access to three python scripts, and compare the performance of the clusters to running the scripts on a single processor. These scripts are available on GitHub[12]. Students will go on to use different versions of the scripts when learning about multi-threading on their own machines.

There have been two main challenges in developing the infrastructure for the project. The first is providing access to the clusters, from a distance, in a secure manner. The OpenSTEM Lab has experience in doing this for other types of experiments, but developing the expertise to do this with the Raspberry

[8] https://coderunner.org.nz/

[9] https://projects.raspberrypi.org/en/projects/build-an-octapi

[10] http://stem.open.ac.uk/study/openstem-labs

[11] This deployment has been significantly disrupted by the Covid-19 situation.

[12] https://github.com/dg7692/TM129

Pi platform has taken longer than anticipated. This has slowed this deployment to students.

The other main challenge was the release of the Raspberry Pi 4. The initial development was for three Raspberry Pi 3 clusters. Hence, further design and testing was required o ensure that the cluster design was suitable for the new hardware. The physical construction remained consistent, but a new codebase needed to be developed. Over time, the the Raspberry Pi 3 clusters will be replaced with newer Raspberry Pi 4 clusters.

Primary impact of this development and study has been providing a broader high quality education to students; without this project students would have limited understanding of distributed computing. This is a significant, given the rapid growth in the Cloud, Edge and Fog computing industries. Learning about these topics in a practical manner makes these students potentially more prepared for the workplace.

On a secondary level, the activities surrounding the clusters take a problem-solving approach, encouraging the students to think creatively; a skill set which can lead through to the workplace.

It is anticipate that the clusters will help improve the retention of students within the Computing and Communications curriculum, thus helping produce more graduates with technical skills.

The ability to access complex and expensive hardware remotely enables us to be open up the course, and the technology, to a more diverse group of learners that may not, for fiscal or other reasons, otherwise be able to access clustered processing capability.

2.4 Interactive textbooks as Jupyter notebooks

The second year Open University module M269[13] (Algorithms, data structures and computability) ends its current life in the 2020/21 academic year. This project is the authoring of a new textbook for the October 2021 start, with an increased focus on problem-solving and programming, in part to address the forthcoming change of student demographics, as more Mathematics students must take M269, a core module in the Data Science BSc[14].

The textbook will be a collection of Jupyter notebooks for students to download from the VLE. No part of the textbook will be available in print or as VLE pages.

Jupyter notebooks are interactive multimedia web-based files. They include both text and code. This makes them ideal for Computing textbooks, in which code can be interwoven with text explaining the algorithms and data structures used in the code. Students can code directly in the notebooks, without constantly switching between a text-only book and a coding environment. This lowers the barrier to programming practice (which is essential for a module like M269) and makes study much more efficient for our time-poor mature students, many of whom with a full-time job.

[13]http://www.open.ac.uk/courses/modules/m269

[14]http://www.open.ac.uk/courses/statistics/degrees/bsc-data-science-r38

Jupyter provides a reasonable coding environment, but isn't a good text authoring environment. The writing the book uses Markdown (which is the text format for notebooks), using the Atom editor and its Hydrogen plugin. This provides a Jupyter-like environment with a proper Markdown editor. A Python script has been developed that generates the notebooks from the Markdown files (one file/notebook per section). The script also numbers headings, creates previous/next links at the bottom of each notebook, and extracts all code from the notebooks into individual Python files for students who need to use a different coding environment for accessibility reasons.

nbsphinx[15] is used to generate alternative formats (PDF, ePUB and HTML) to make it easier to search for particular terms across the whole book.

Students will need to install Anaconda[16], which includes Python, Jupyter and other libraries needed for M269. However, it is likely that a Jupyter server will be provided, building on the development for TM351[17] (Data management and analysis), for students who are unable to install Anaconda.

It is yet to be seen if nbgrader[18] (a tool for marking assignments written as notebooks) can be integrated with the rest of the OU's assessment infrastructure.

Naturally, when working with five different formats, there are several incompatibilities or mismatches. For example, lists in notebooks use letters (a., b., ...) while the PDF has numbers (1., 2., ...). The ePUB file doesn't render math notation or framed text correctly. The current focus is on producing content, so formatting issues will be dealt with later. Some may be fixed with appropriate CSS styling, others may remain unsolved.

Overall, the approach is working quite well to support diversity and inclusive and potentially scalable approach. The PDF and HTML generated by nbsphinx with minimal configuration provide good visual outputs that can be easily navigated. An ePUB is likely not the best format for a Computing textbook.

Jupyter and Anaconda are widely used in the data science community, preparing students for the world of work. Other modules in the Data Science BSc currently in production will also use notebooks, but not to the same degree as M269.

Several M269 exercises are adapted from competitive coding platforms such as Kattis[19]. Many employers use similar kinds of problems in their technical interviews.

2.5 Badged Course: Digital Thinking Tools

In this project, a team at the Open University authored a badged course[20] on "Digital Thinking Tools"[21]. This free course starts by placing thinking tools

[15] https://nbsphinx.readthedocs.io/en/0.6.1/
[16] https://www.anaconda.com/
[17] http://www.open.ac.uk/courses/modules/tm351
[18] https://nbgrader.readthedocs.io/en/stable/
[19] https://open.kattis.com/
[20] https://www.open.edu/openlearn/get-started/badges-come-openlearn
[21] http://open.ac.uk/digital-thinking

in their historical context and concludes with an examination of their future. During the course, students are introduced to a wide variety of digital thinking tools. They apply these tools in many practical activities and case studies, solving problems that involve finding and evaluating information, performing calculations and drawing reasoned conclusions.

The course will appeal to anyone who uses information to answer questions, solve problems or make decisions, whether it is in their personal or professional life, or as part of their studies. It introduces a range of digital thinking tools that can help with finding and analysing information, and reasoning with that information. Students will, among other things, learn how to become a 'super-Googler', use their computer as a big calculator (with the Python programming language) and make digital argument maps that shed light on the reasoning behind their own writing and that of others. The course concludes with an overview of the history, pitfalls and prospects of Artificial Intelligence.

The course aims to navigate between appealing to a wide audience (with its approachable examples and eight 3-hour study blocks) and introducing students to real practical tools and challenging skills (from constructing and understanding complex arguments to using Python to solve Fermi problems). The free 24 hour online course, which can be taken anytime, anywhere, went live on February 6, 2020. Two month later, there was approximately 1,800 active registered learners and 20,000 course views.

This course helps students approach problems in novel ways. It encourages both creativity and systematic thinking, which can lead through to the workplace.

Successful completion of this course (which includes two summative quizzes) results in an OpenLearn badge. Currently overr 400 learners have completed the course and over 2000 learners are currently actively engaged. The course has been endorsed as Bronze Level knowledge by the Institute of Coding. Once a student has earned their OpenLearn digital badge, they can use this badge to also claim their Institute of Coding badge on the 'Claim a Badge' page. It is also worth noting that this course has been endorsed by the Society for the Study of Artificial Intelligence and the Simulation of Behaviour [22] as part of its mission to support the public understanding of AI.

2.6 Challenges with learning to program and problem solve: an analysis of students' online discussions

Students who study problem solving and programming (in a language such as Python[23]) at University level encounter a range of challenges, from low-level issues with code that won't execute or compile to misconceptions about the threshold concepts and skills. The current project complements existing findings on errors, misconceptions, difficulties and challenges obtained from students after-the-fact through instruments such as questionnaires and interviews. In

[22] https://aisb.org.uk/87-aisb/publicrelations/93-mutual
[23] https://www.python.org/

our study, we analysed the posts from students of a large cohort (1,500) of first-year University distance learning students to an online 'Python help forum' - recording issues and discussions as the students encountered specific challenges. Posts were coded in terms of topics, and subsequently thematically grouped into python-related, problem solving/generic programming related, and module specific.

The final report for this project documents the full set of topics and the statistics for each of them. It also provides examples from the forum discussions which illustrate the topics that were identified. The software we developed for analysing Moodle forum posts has been made available via the Open University's open resources platform. The work was presented as a paper at the 51st ACM Technical Symposium on Computer Science Education (SIGCSE 2020[24]). A video recording of the presentation is also available.

In this project, the study succeeded in analysing one of the, to our knowledge, largest collections to date of University student discussions around problem solving and programming challenges (1,500 forum posts). To deal with this number of Moodle postings, the design and development Python3 code solution was implemented to automate data collection from the forums.

This work provides educators with insights into areas where beginning University students face challenges with learning to problem solve and program. This should help develop higher quality education in this area, which can, in turn, lead to learners who are more prepared for the workplace.

The work in this project contributes to the Institute of Coding Observatory's aim of a forward-looking research community developing an outstanding evidence base of knowledge and best practice for digital skills needs and training.

2.7 Are You Ready for TM351? A Moodle diagnostic quiz

The goal of the project was to design a quiz to give level 2 or other students preparing for the level 3 module TM351[25], 'Data Management and Analysis' an idea of the level of Python programming they would need to be able to access the module material.

Three groups of questions, graded Easy, Intermediate, and Hard were developed, using mainly CodeRunner[26] questions integrated into the Open University Moodle Learning Management System.

The easy questions were at a level which most second-year students of computer science would be able to manage, perhaps looking up any syntax needed if they are unfamiliar with the language (as they can do throughout their study). For example: *how many items in a list?*.

The intermediate questions were designed to reflect the minimum familiarity and skill to get started on the module materials. For example: *remove punctuation and white space from a string, returning only alphabetic characters.*

[24]https://sigcse2020.sigcse.org/
[25]http://www.open.ac.uk/courses/modules/tm351
[26]https://coderunner.org.nz/

The hard questions go beyond the expectations and to solve them using only standard library functions is not taught on the module, which uses specialist libraries. Any student who can cope with these can be very confident that they are ready.

CodeRunner was used to design, develop and test the quiz. Other modules use the same approach for 'Are You Ready For' diagnostic self-assessment quizzes. It is a widely used tool to support self-paced preparation and learning through assessment to help students prepare for the start of a module.

TM351 uses Jupyter[27] Notebooks and specialised code libraries such as pandas[28] but assumes that these are new to students: the goal of the quiz was to assess confident use of Python to solve simple computational tasks and to some extent to deal with small data-sets.

Python is taught as a means to the end of being able to conduct the investigation and analysis of large real-world data-sets such as UK crime, arrest and conviction statistics. Students who identify their readiness might feel confident that they are ready to apply themselves to the module, or perhaps follow prompts to revise or improve their skills before starting.

The ethos of the Open University has embraced diversity and inclusivity as founding principles for fifty years. The willingness to welcome students with a wide range of precursor knowledge and experience is exemplified by the support offered through the provision of preparedness quizzes as demonstrated in this case study.

2.8 Industry mentoring programme for women Computing and IT students (Pilot)

The Open University Industry Mentoring Programme (IMP) is a pilot project that aimed to support women Computing and IT students in their career progression, through mentoring by experienced individuals who are already working in the Computing and IT industry. The focus of the mentoring relationship was on supporting the mentee to develop their career ideas, clarify goals and identify pathways to achieve those goals, with the aim of improving employability.

The project aimed to:

- Recruit 20 women students from 3rd level Computing & IT modules as mentees.
- Recruit mentors, preferably Open University graduates, who are working in IT industries.
- Carefully match mentees with mentors.
- Support the mentoring relationships throughout the programme.

[27] https://jupyter.org/
[28] https://pandas.pydata.org/

The project used the Open University's database of students to identify women students on four Level 3 Computing & IT modules and obtain an email address. These students were sent an email invitation to become a mentee. Respondents were sent further information about the project and a Mentee-Mentor Matching form to complete and return.

In order to recruit mentors, the Open University's Alumni Association[29] agreed to identify alumni who had graduated in the previous 10 years with the B62/Q62 BSc in Computing and IT or the B67/Q67 BSc Computing and IT and another subject. In addition, the net was widened using personal links and social media, including the Open University Careers Network (TOUCaN) LinkedIn group[30] and the School of Computing & Communications' Industry Partnership Board. Those who responded were sent further information about the project and a Mentor-Mentee Matching form.

The mentees were matched with mentors using information provided in the Matching forms, such as the mentee's goals and aspirations and the mentor's areas of expertise, with the Project Consultant supporting a process of agreement between mentor and mentee.

Training and support was provided for participants using the following resources:

- Online briefing sessions using Adobe Connect for mentors and mentees separately to introduce the programme.
- Input from Careers and Employability Services to the mentee briefing session.
- OpenLearn course: Exploring career mentoring and coaching[31].
- A final online meeting in Adobe Connect open to both mentors and mentees, as part of closing stages of the mentoring process.

Recordings of the online introductory briefing session for mentees and the final meeting for all participants were made available to all participants as MP4 files in a public DropBox folder.

Choice of communication method between mentors and mentees was left to their own preferences. Participants reported communication via email, text message, WhatsApp and LinkedIn.

There was a great deal of enthusiasm and excitement about the project from both mentors and mentees.

There were 21 positive responses from the female students contacted on the first four modules included in the project. However, the biggest challenge was in the recruitment of mentors. The project had intended to recruit both mentors and mentees alongside each other so to make the most appropriate matching. Unfortunately, recruitment of mentors started later than expected

[29] https://alumni.open.ac.uk/
[30] https://www.linkedin.com/groups/3871260
[31] https://www.open.edu/openlearn/money-business/available-now-exploring-career-mentoring-and-coaching

and only five mentors were recruited from amongst the alumni. Therefore, mentee recruitment was stopped until sufficient mentors were recruited for those already recruited.

Although sufficient mentors were recruited eventually through personal contacts and social media, the time interval created practical difficulties, for example, in organising a briefing session for mentors, and some mentees may have lost interest in the project as they did not respond to emails after a time, or the mentoring partnership foundered early on. Another challenge was that two potential mentees expressed preference for female mentors as role models, but there were not enough female mentors available to support them.

The final Adobe Connect meeting was used to elicit feedback from participants. They reported a wide variation in experiences ranging from some very successful mentoring pairings that intended to continue for several months beyond the life of the project to one pairing where the mentee had not responded, and others with varying levels of contact.

The aim of the project was to promote employability through mentoring. Some mentors have commented that their mentees had limited awareness of how their Open University qualifications might help them find work, and they had little or no awareness of how to use professional social networks, such as Linkedin, so the mentor could help with this. Mentees have commented on the importance of being able to communicate with an industry professional as shown in the comments from Progress Updates.

"... I was given a chance to think realistically on how should I approach my dream career." (Mentee 1)

"I have had some very productive email "conversations" with my mentor helping to establish the direction I would like my career to take. This has come at a crucial time when I have the opportunity to take a secondment within my current organisation and my mentor has been very helpful in talking about some of his experiences as well as helping me to reflect on my own skills and possible options available to me." (Mentee 3)

Digital badges from the IoC will be offered to mentors and mentees.

2.9 Ethics for Cyber Security

Understanding ethics in the context of information systems has been an on-going challenge and study for many decades. As information technology and advances of these technologies become more ubiquitous in society, the influence, bias and ethical concerns around IT affordances have increased. Teaching and learning about Ethics and its importance in design, development and use in IT systems is complex. To help bridge this challenge and support the engagement with learners from a diverse background the research and study into use of dialogues, a form of narrative pedagogy has been investigated.

Dialogues are an attempt to engage small groups of learners in playful discussions of ethical dilemmas. Each dialogue consists of a simple scenario which is made increasingly complex, allowing players to reconsider their assumptions and change their opinion based on new information.

The dialogues take the form of illustrated cards and associated teaching instructions. A dialogue begins by laying down a card outlining the initial scenario. Further cards detail potential consequences from the scenario. One such scenario is the so-called 'trolley problem'[32] developed by the philosopher Philippa Foot. A summary of the problem is as follows: 'A runaway streetcar is rolling down a street towards a group of five people. A set of points is located between the trolley and the people which can switch the trolley on to another track – but there is one person on this track. You have control of a lever operating the points. If you do nothing, the trolley will hit all five people; if you choose to operate the switch, the trolley will be diverted and hit one person. What do you do?'

The group discusses the initial scenario, explores the consequences of various actions (or in-actions) and attempts to determine a policy acceptable to all of the members. The scenario is then expanded with one or more cards which serve to complicate matters or give additional information that was not present in the original scenario. For instance the trolley problem has been enhanced in a number of ways such as by being able to stop the trolley by pushing a bystander in front of the train, by placing different values on lives – such as some of the participants being old, sick or criminals. These complications serve to challenge some of the assumptions made in the original scenario and force participants to confront the morality of decision-making.

Each dialogue ends with a short discussion of why the scenario is relevant to the real world. The trolley problem is widely-cited in discussions of the ethics behind the development and deployment of autonomous vehicles in which software designers will have to develop strategies for dealing with accidents involving passengers and bystanders. The dialogue serves to introduce ethical issues to learners which can then be followed up in other activities, such as researching topics raised during the dialogue or reflecting on their experience of the dialogue.

The dialogues demonstrate that ethical considerations are fundamental whenever humans interact with technology and those responsible for designing and deploying novel technology should be obliged to actively engage with the ethics of their decisions. The dialogues also aim to show that even the simplest scenarios may have no perfect solution.

2.10 AI Workshops

Part of cross theme engagement between theme 1, 3 and 4 was the development of an AI and Machine learning summer school. The aim of the summer school was to engage with local communities, SMEs and public from a diverse background to assist in demystifying AI.

The summer school designed to provide introductory materials about the concepts of AI and machine learning. Also, practical activities from getting started and applying key concepts of machine learning were designed. Although

[32] https://en.wikipedia.org/wiki/Trolley_problem

the programming language used was Python, no programming knowledge of the participants was expected and each step by step activity was explained and supported by a number of academics, tutors and PhD students.

The resources are available GitHub[33] as a code repository and Google's Colab[34] as a notebook engine was used. While similar to Jupyter notebooks, Colab was chosen as it requires less configuration and is able to provide consistent libraries and access to processing power that may have been an issue had we attempted to get the (unknown) attendees to install Jupyter on their own machines.

The hands-on sessions were on consecutive days with the first day being focused on image recognition via neural networks and the second day being time series classification. Both sessions were split into a number of sections, each with their own notebooks. The attendees followed through the notebooks (as lead by the host of the session), ran the provided code, and were then given the opportunity to modify parameters within the code and examine the changes in the results.

With no prior knowledge required, there was a vast range of abilities in the room, from a mature learner that claimed to know virtually nothing about computers, to a sixth-form Computer Science student, to IT professionals. This meant a variety of support was needed. The use of pre-configured cloud-based technology reduced the impact of the technological ability range. Using notebooks with explanation and plenty of extension activity in each one meant those who struggled with verbal guidance were supported in the initial phase, and those who were more confident were able to carry on learning. This left the facilitators free to deal with more serious misunderstandings. This also meant that the learners could revisit or extend their learning outside of the workshop at their leisure.

Machine Learning is a growth industry predicted to be worth US$20.83B (£16.79B) by 2024[35]. While there are degree courses which cover the topics of machine learning and artificial intelligence, gaining access to expert training outside of a full course is challenging and the workshop was a popular and well received opportunity. Furthermore the use of Colab notebooks, which are similar in nature to the Jupyter notebooks discussed elsewhere, proved popular with the learners and were an effective way of freeing up the facilitators. Colab was designed to meet the business needs of Google and while there are no statistics, the increasing use of it and similar tools such as Jupyter, as well as related technologies such as TensorFlow[36] (also covered in the workshop), suggest that the technologies are becoming the de-facto standards within the industry.

The workshop had a diverse range of attendees of both genders, a wide range of ages, and as mentioned above, a significant difference in ability. This diversity ties directly in with the underpinning aims of both the IoC and the Open

[33]https://github.com/

[34]https://colab.research.google.com/

[35]https://www.forbes.com/sites/louiscolumbus/2020/01/19/roundup-of-machine-learning-forecasts-and-market-estimates-2020/

[36]https://www.tensorflow.org/

University. The summer school also experimented with the idea of combining digital badges with physical badges. As each learner completed an activity they received an open badge and the physical badge was updated to indicate the change. The physical badge had been designed by the OU team to support further digital skills development. This offered an innovative bridging of learning between digital and physical developments. This idea was further developed and piloted at the IoC conference in 2020.

A further output of sharing these resources and engaging with SME communities has been the support in the development and engagement with IoC project. Also, the resources are being used by some of the SME communities to help in the development of their data science projects. These resources have been used to inspire further teaching and learning across communities and the design of the summer school has been used as a template for further engagement with other networks, such as SERAS https://seras.org.uk.

2.11 'Cisco NetAcad' - Teaching the Teachers

The OU, in collaboration with the UK extension of the global Cisco 'NetAcad' CSR programme[37], recognised that a primary cause of the national digital skills short focusses on the skills deficit amongst many secondary computing school teachers and further education lecturers. Often, their skills remain static in an ever changing employability demand domain and with demanding jobs and external compliance pressures – seldom have time to maintain their skill set.

Focussed around Network Engineering, CyberSecurity, Computer Systems Technologies, Linux and Python. The OU Cisco ASC (academy support centre) formed a focused collaboration under the umbrella of the IoC to offer a Hybrid-MOOC taking elements of the Cisco NetAcad VLE and other engagement tools. Presenting a 100% remote teaching experience that has to date, reached 5000+ teachers and accredited (in Cisco terms) over 1800+.

Following the 'mashup' principle (Minhas et al. 2012), the OUCisco team focussed on creating a bespoke mashup of specific engagement tools. Adapting the NetAcad Canvas resource and focussing on key delivery methods. This was accomplished by through a number of approaches; We created a focussed community within social media, namely Facebook[38] – inviting participants via various channels to join this page. The community was seeded with pre-prepared knowledge based posts, set around the subjects being covered in line with each teaching sequence. Weekly live webinars were hosted, tied in with the study pattern of each MOOC session. These were curated into playlists and fed into the NetAcad platform. A 'catch-up TV' collection of webinar sessions were maintained. Volunteers were solicited to support the community of teachers from the wider community of Cisco Academy educators.

We also hacked the NetAcad Canvas features to operate as an integrated MOOC (NetAcad is typically designed for a Face-To-Face experience, the OU

[37] https://www.netacad.com/
[38] https://www.facebook.com/OUCiscoASC/

have adapted this over 15 years of experience). The self-enrolment features of lower level courses within NetAcad were leveraged to act as enrolment/import gateways into more advanced courses being offered to teachers. Content was pointed to the Facebook Live sessions and associated Facebook page. Clear study planner guidance was added and MailChimp[39] emails were syndicated into the course.

The free affordances of MailChimp were leveraged following a behavioural science (nudge theory) principle (Ebert & Freibichler 2017), for reminding students, focussing them on assessment thresholds and managing population performance. Emails were linked to weekly Facebook Live broadcasts. Students were directed to key points within content and provided with study tips following the study plan. Encouragement (nudges) were given around key assessment activities.

This project is an iterative successor of previous Cisco 'instructor' outreach projects led by the OUCisco team. In some respects, many of the mistakes were made before the IoC project went live. New challenges that occurred include:

1. Discovering the affordance changed at 1000 – having to change the assessment activation methodology (Cisco were also surprised by this.
2. Encouraging some students to focus on the discussion forums – not, replying to emails. With modified nudges/coaching, this worked and was an iterated improvement during the life of each subsequent course. More work was done to direct students to the correct platform tools.
3. Setting cross site cookies as NetAcad is a collection of platforms – some students assumed the error message was a system issue, not a local browser issue.

There were, however, some significant successes including:

1. Improved retention – a typical MOOC enjoys 5 to 10% - during our run with Cisco prior to the IoC project we enjoyed 22 to 25% retention/achievement. With refinements, focus of content – nudges and interactions this increased to 28% to 35%.
2. Scale – one tutor+admin scaling equally for 1, 100 or 1000 participants at the same cost per presentation.

The primary focus of this project working with Cisco was to improve the advanced technological skills of school teachers and further education college lecturers. The Cisco accreditation is inherently focused on global sector oriented industrial readiness. The 1800+ teachers are now able to access advanced content and technical resources and pass onto their students. We are already seeing a 30%+ participation up to 11500+ students this year and growing.

While the target audience of the project has been restricted to teachers, the courses attract a diverse range of attendees in terms of gender, ability, prior learning and career position.

[39] https://mailchimp.com/

3 Recommendations

The case studies included within this report describe a wide range of projects that have been carried out within the Open University and either under the Institute of Coding banner, or in conjunction with Institute of Coding projects. A variety of recommendations can be extrapolated both as specific actions and generalities.

While the benefits of pair programming have been widely reported (Smith et al. 2018) the current research is primarily focusing on programmers who are physically co-located; the ongoing investigation in case study 2.1 shows early suggestions of benefits when used by programmers who are remote to one another, a situation made far more prevalent by the COVID-19 pandemic. Any remote-teaching establishment could be recommended to monitor the study and consider the use in their programming courses.

The use of Jupyter as both an infrastructure as in case study 2.2 and a pedagogical approach as in case study 2.4 has proven a popular area of investigation and follows on from the preceding report. The benefits, while particularly of interest to remote learners and the Open University model of teaching, would be just as useful more more traditional approaches. The infrastructure allows for students to use the same environments at home as in labs and the pedagogical approach of using a notebook-based textbook is of equal benefit across an increasing range of courses. The redevelopment of the popular Open University module M269 has all of the course material being developed in Jupyter notebooks. In addition, the TMAs (tutor marked assessments) which will be written, completed, and assessed entirely in Jupyter notebooks. The modules TM351 and TM129 are both also primarily Jupyter based, running on docker[40] containers. The TM129 refresh[41] uses Jupyter notebooks to integrate a novel robot simulator[42].

Distributed computing is not a new challenge in Computer Science education (Paprzycki et al. 1995, Paprzycki 2006) and while some of the difficulties are mitigated by the prevalence of multi-core processors, access to cluster-computers is still a problem for face to face institutions, and even more so for remote learning institutions such as the Open University. The novel approach demonstrated in case study 2.3 would be reproducible by both kinds of institutions and demonstrates an approach which is both effective and engaging.

The three case studies 'Digital Thinking Tools' (case study 2.5), 'Analysis of students' online discussions' (case study 2.6) and 'Are You Ready for TM351?' (case study 2.7) all demonstrate creative ways of supporting student learning. The addition of a digital badge, as used by case study 2.5, is a way of permanently recognising learning and achievement as well as bolstering motivation (Gibson et al. 2015). This is again demonstrated in the mentoring programme (case study 2.8) which supports a key group of students and an innovative mix of digital and physical badges was trialled at the Artificial Intelligence workshop

[40]https://www.docker.com/
[41]https://tinyurl.com/y5wm4ou3
[42]https://tinyurl.com/y2ub4tsg

(case study 2.10)

The authors of 'challenges with learning to program and problem solve' (case study 2.6) suggested in Savage & Piwek (2019) that:

> Students that begin to learn to program can be supported with wide variety of tools, e.g. for editing their code, marking it and tracing its execution. Deployment of such tools should however be undertaken with care, since adoption and use of the tools themselves can interfere with the student's learning and cause them to focus on issues with the tool use rather than the computing and programming concepts they are trying to master.

and that:

> Students that are learning to program should be provided with support on how to express problems with their code. This could include explicit guidance on:
>
> - Including the code they tried to execute.
> - Describing the behaviour/output they expected from the code.
> - Describing the actual behaviour and output.

Further discussion can be found in Piwek & Savage (2020).

The final case study 2.11, that is part of theme 2 work provided significant impact and results for teaching the teachers. Part of the success has resulted from the reach (using social media) and flexibility of learning that the Cisco platform offers. But even more than this has been the authentic learning to practice of the resources, that is the resources the teachers learner can be immediately used to deliver to the students. There are clearly other key design factors, such as gaining a recognised certification opportunity and that the resources are always up-to-date.

Novel technologies both within the Open University, and in the field of Computer Science at large; Some of the challenges are discussed and these inevitably lead to ethical implications which are often ignored or not tackled effectively (Gotterbarn & Miller 2018). The discussions highlighted in case study 2.9 go some way to challenging these circumstances.

4 Contributors

The case study in section 2.1 "Pair Programming for Students Learning Programming at a Distance" was contributed by Adeola Adeliyi

The case study in section 2.2 "Jupyter Infrastructure " was contributed by Alistair Willis, Tony Hirst and Patricia Charlton.

The case study in section 2.3 "Teaching distributed computing using Raspberry Pi clusters at a distance" was contributed by Daniel Gooch, Jon Rosewell and Mike Richards.

The case study in section 2.4 ”Interactive textbooks as Jupyter notebooks” was contributed by Michel Wemelinger

The case study in section 2.5 ”Badged course: Digital Thinking Tools” was contributed by Paul Piwek (Academic lead) Richard Walker (Consultant).

The case study in section 2.6 ”Challenges with learning to program and problem solve: an analysis of students' online discussions” was contributed by Paul Piwek (lead) and Simon Savage (Consultant).

The case study in section 2.7 ”Are You Ready for TM351? A Moodle diagnostic quiz” was contributed by Charly Lowndes.

The case study in seecion 2.8 ”Industry mentoring programme for women Computing and IT students (Pilot)” was contributed by Elaine Thomas (Project leader) and Geraldine Doyle (consultant).

The case study in section 2.9 ”Ethical dialogues” was contributed by Mike Richards.

The case study in section 2.10 ”AI Workshops” was contributed by Oli Howson and Patricia Charlton.

The case study in section 2.11 ”'Cisco NetAcad' - Teaching the Teachers” was contributed by Andrew Smith.

5 References

Celepkolu, M. & Boyer, K. E. (2018), 'The importance of producing shared code through pair programming', *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education* **2018-Janua**, 765–770.

Ebert, P. & Freibichler, W. (2017), 'Nudge management: applying behavioural science to increase knowledge worker productivity', *Journal of Organization Design* **6**(1), 1–6.

Ghorashi, S. & Jensen, C. (2016), Supporting Learners in Online Courses Through Pair Programming and Live Coding, *in* '2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)', Atlanta, GA, pp. 738–747.

Gibson, D., Ostashewski, N., Flintoff, K., Grant, S. & Knight, E. (2015), 'Digital badges in education', *Education and Information Technologies* **20**(2), 403–410.

Gotterbarn, D. & Miller, K. (2018), 'The Continual Evolution of Interest in Computing Ethics', **9**(2).
URL: *https://twitter.com/cfiesler/status/9312005758734*

Minhas, S. S., Sampaio, P. & Mehandjiev, N. (2012), A framework for the evaluation of mashup tools, *in* 'Proceedings - 2012 IEEE 9th International Conference on Services Computing, SCC 2012', IEEE, pp. 431–438.

Paprzycki, M. (2006), 'Education: Integrating parallel and distributed computing in computer science curricula', *IEEE Distributed Systems Online* **7**(2), 45–50.

Paprzycki, M., Wasniowski, R. & Zalewski, J. (1995), Parallel and distributed computing education: A software engineering approach., *in* R. Ibrahim, ed., 'Software Engineering Education.', 895 edn, Springr, Berlin.

Piwek, P. & Savage, S. (2020), Challenges with Learning to Program and Problem Solve: An Analysis of Student Online Discussions Conference or Workshop Item Challenges with Learning to Program and Problem Solve: An Analysis of Student Online Discussions, *in* 'The 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)'.
URL: *http://dx.doi.org/doi:10.1145/3328778.3366838*

Savage, S. & Piwek, P. (2019), Full report on challenges with learning to program and problem solve: an analysis of first year undergraduate Open University distance learning students' online discussions, Technical report.

Smith, M. O., Giugliano, A. & DeOrio, A. (2018), 'Long Term Effects of Pair Programming', *IEEEE Transactions on Education* **61**(3), 187–194.