Vol.8 No.1 (2025)

Journal of Applied Learning & Teaching

ISSN : 2591-801X

Content Available at : http://journals.sfu.ca/jalt/index.php/jalt/index

Edubuntu: The overhaul and rebirth of an educational operating system

Poramate Tarasak[A] A *Institute for Innovative Learning, Mahidol University, Thailand*

Curtis J. Bonk[B] B *School of Education, Indiana University, U.S.A.*

Suthiporn Sajjapanroj[C] C *Institute for Innovative Learning, Mahidol University, Thailand*

DOI: https://doi.org/10.37074/jalt.2025.8.1.29

Abstract

Computer operating systems are often an overlooked part in providing quality education. When discussing digital education tools, most practitioners and researchers mainly focus on hardware (e.g. PC, notebook, tablet, and smartphone) and software applications that directly interact with students. While Microsoft Windows is widely recognized, other alternatives remain relatively unknown. This paper introduces an alternative approach by examining the potential of open-source operating systems such as Linux, particularly focusing on Edubuntu, a flavour of Ubuntu operating system, designed specifically for education. It also explores available default applications and investigates how Edubuntu can provide expansive learning opportunities that develop learners' interest and help cultivate 21st century skills. Importantly, we pose the question of how to harness the potential of Edubuntu potential effectively. We then illuminate Edubuntu's capacity to support traditional subjects as well as promote computational thinking and geographic education. Lastly, we analyze Edubuntu with respect to instructional software design standards and summarize key consideration factors in designing an educational operating system.

Keywords: Edubuntu; educational needs; educational operating system; open source software for education; sustainable development; teaching computational thinking; teaching geography.

Introduction

In efforts to utilize digital technology for bridging educational gaps, various global initiatives have been implemented. One of the most famous projects is perhaps One Laptop Per Child (OLPC) from MIT, whose mission was to provide low-cost laptop technology to underprivileged children as a learning tool (OLPC, 2005). Although the project was deemed unsuccessful (Ames, 2021), it was a valuable lesson in hardware and software design for a minimal-cost computer. The system design attempted to produce a laptop for under USD$100, whose functionality was solely to support the education of young children. An OLPC laptop had its own desktop environment called Sugar, running on the free Linux operating system (OS). As such, it was packed with free open-source software (OSS) for learning subjects such as science, mathematics, programming and music (OLPC, 2013).

To achieve the UN sustainable goal under the quality education category, OSS and Linux are perhaps indispensable components of any educational computing project and initiative due to such software providing free license of use as well as free distribution in nature (Pan & Bonk, 2007a, 2007b). There was debate whether to adopt OSS to teach children, as it may not be beneficial in the real-world employment settings (van Rooij, 2019). However, technological advancement and the proliferation of diverse devices and platforms, it is crucial for learners to gain experience and understanding across multiple platforms. Increased exposure and familiarity with various tools and technologies enhance adaptability to new ones, aligning with highly valued 21st century skill sets sought by employers.

A systematic review of 69 research studies showed that OSS use was challenged by poor documentation, limited functionality, lack of awareness, and a general lack of OSS knowledge and programming skills (Talib et al., 2019). Their follow-up study with 400 school teachers in the United Arab Emirates (UAE) found that these teachers faced many barriers to integration and use of OSS, including internet access as well as sufficient time and organizational support needed to integrate OSS in computing courses at the high school level (Talib et al., 2019).

The difficulties in applying OSS in education are well-recognized (Özüsağlam & Poşpoş, 2016). One reason is the lack of support from IT staff and the absence of easy-to-understand online documentation and tutorials. Another problem is the constant barrage of new open course technology tools and opportunities (Pan & Bonk, 2007a, 2007b), as well as the short-lived nature of many, if not most, of those tools and systems (Butler et al., 2022). Nevertheless, recent trends in using community support resources such as stackoverflow.com, askubuntu.com, or social media groups are increasingly becoming the norm for any internet users. The ability to tap into such gigantic knowledge banks and connections at any moment will gradually mitigate the fear of OSS adoption across educational sectors and discipline areas, especially, when artificial intelligence (AI) assistants have advanced to the point that masses of people are comfortable using them (Rudolph et al., 2023) and depend on them (Huang et al., 2023; Hwang et al., 2020).

The new era of educational technology has made online learning activities and tools a ubiquitous part of one's daily routine and lifelong learning opportunities. The general OS functional requirement of simple and efficient resource management is not sufficient for a special-purpose usage such as education (Hansen, 1973). A well-designed OS for education should be lightweight and feature a simple and beautiful interface. Being lightweight and having small memory/CPU footprint features allow the OS to be installed on any machines or even just on a thumb drive. Previous criticisms pointed out that using technology for education effectively goes beyond being a mere tool (Ames, 2021; Souter, 2021). Students cannot learn on their own without the guidance of teachers or assistance with technical issues related to the use of laptops, tablets, and other technology devices. Therefore, an ideal educational OS should also include an ample selection of pre-installed software that allows students to work independently with minimal need for teachers' technical expertise to support them. In this regard, there are many existing educational OSs to consider. A feature comparison of selected OSs is presented in Table 1.

Table 1: Educational OS feature comparisons.

OS name	Linux Distribution	Student Levels	Teaching Subjects	Status
Edubuntu	Ubuntu	Primary to Higher education	Various	Active
UberStudent	Ubuntu	Secondary	Various	Active
Qimo	Ubuntu	Early Childhood to Primary	Math, Drawing, Computer Skills, Brain Training	Discontinued
AcademiX	Debian	Primary to Higher education	Various	Active
Debianedu/Skolelinux	Debian	Primary to Secondary	Various	Active
Sugar	Many	Early Childhood to Primary	Arts, Math, Physics, Brain Training	Active
MentOS	Many	Higher Education	Operating System	Active

Since Ubuntu is the most popular Linux distribution and Edubuntu is promoted officially on the Ubuntu flavor website (Canonical Ltd, 2024), we selected Edubuntu to investigate its functionalities and applications.

Edubuntu operating system

Edubuntu was first released in 2005 based on the Ubuntu Linux distribution (Edubuntu, 2024). It was developed in cooperation with teachers from many countries and originally targeted users aged from six to 18. Importantly, Edubuntu has been instrumental in expanding learning opportunities and cultivating 21st century skills. For instance, in India, it was utilized to teach computer science through innovative curricula like Computer Masti, leveraging applications such as GCompris, Childs Play, Tux Paint, Tux Math, KDE Educational Games, and Scratch (Baru et al. 2009). Similarly, in the Republic of Macedonia, the Bureau for Development of Education specified Edubuntu to be taught in the following subjects: Computer Science, Mathematics, Physics, Chemistry, Geography, Music, and Latin (Barbareev, 2015). Notably, in that initiative, teachers received formal training before introducing Edubuntu to the students. Several other countries also have used Edubuntu in schools such as the U.S.A., Belgium, Serbia, Malaysia, and Ghana (Ubuntu Community, 2013).

Despite its initial impact, Edubuntu faced discontinuation in 2016 due to its' lead developers moving to another project (Graber, 2016). Being left unmaintained for about six years until December 2022, Edubuntu found new leadership under Amy Eickmeyer from the United States and her team, who embarked on reviving and enhancing the operating system (Eickmeyer, 2022). The revitalized Edubuntu, which was built on Ubuntu OS release 23.04 Lunar Lobster, boasts easier installation procedures, a new logo, and a revamped website (edubuntu.org), ensuring continued accessibility and relevance in fostering digital literacy. Since then, Edubuntu has undergone a few major change releases, versioning 23.10 Mantic Minotaur, 24.04 Noble Numbat, and 24.10 Oracular Oriole. The review in this paper focuses only on Edubuntu release 24.04 Noble Numbat. Despite it being an older version than 24.10, this release is quite recent and has long-term support for three years (Eickmeyer, 2024).

To provide a comprehensive scope for this study, we focused on Edubuntu's educational features, examining its application in teaching and learning processes and how students can leverage its tools for learning digital technology. Our key research questions are as follows:

1. What are the features and functionalities of Edubuntu, particularly in the context of educational use?

2. What education applications are included by default in Edubuntu, and what are their primary purposes?

3. How can the included educational applications be integrated into teaching and learning processes?

Methodology

Our research takes an exploratory approach. To address RQ1 and RQ2, we conducted an in-depth review of Edubuntu's features, documentation, and user interface. In the process, we identified potential use cases within educational settings

and evaluated the system's effectiveness in meeting educational goals. Additionally, two case examples – computational thinking and geography – were created to answer the RQ3.

Edubuntu 24.04 general specification

In this section, we provide an overview of the fundamental functions of Edubuntu. This detailing of many applications and features available for Edubuntu aims to assist teachers and educators who are considering adopting the platform by outlining the necessary requirements for installation and key considerations such as administrative applications, memory management, user interface, main utilities, and system services.

Installation

To save disk space or to customize to a student level, installations of Edubuntu can target a specific group of students: preschool, primary, secondary or tertiary (shown in Figure 1). It can be installed from an .iso image downloaded from a server or, if the existing OS is already Ubuntu, from the command line '*apt-get install edubuntu-desktop*'. The .iso image file size is 6.2GB, which is beyond DVD capacity, can be written on a USB stick of a larger size for installation.

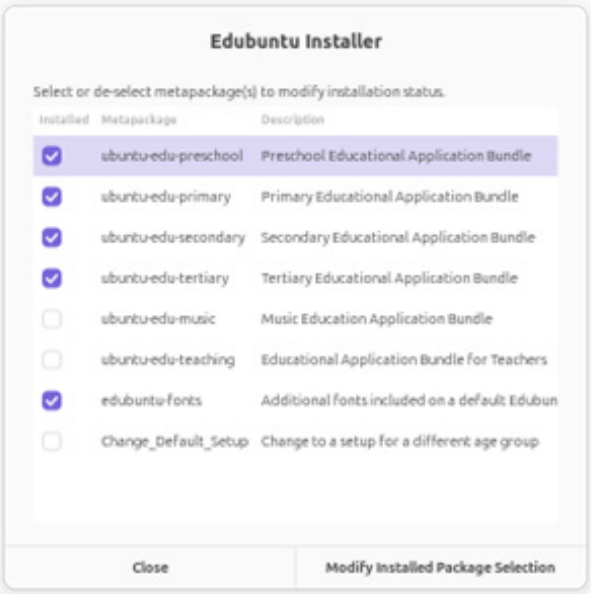

Figure 1: Edubuntu installer.

Apart from installation on a hard disk, Edubuntu can be fully run on a USB stick without installation by booting the PC/notebook with it and selecting the 'Try Edubuntu' (see Figure 2). In this manner, Edubuntu can be easily utilized as a low-cost teaching material and is highly portable. The cost of one cheap USB stick being able to contain Edubuntu is less than USD$2.

It is important to note that the use of Edubuntu and most of its applications (except for the Web browser, mail client, dictionary, maps, and weather applications) do not require internet connections which makes it quite useful in the Global South (Schotgues, 2022; Zhang et al., 2020). In effect, it would be well-suited to teaching/learning in remote areas in the Global North as well.

Figure 2: Installation options.

Administrator

In addition to student tools, there are also administrative applications. Notably, Edubuntu Menu Administrator allows the administrator to hide selected applications from non-administrator users for simplicity of use (see Figure 3).

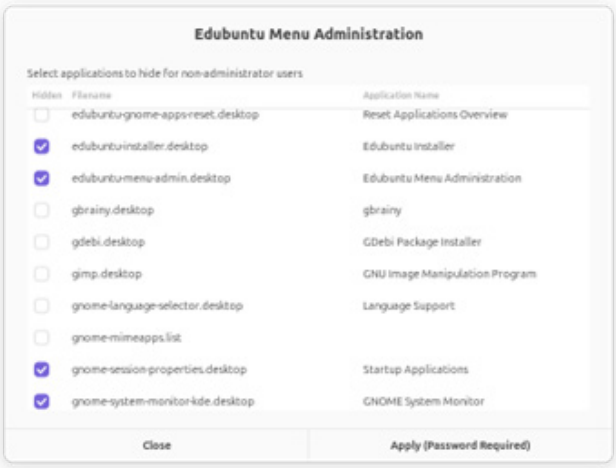

Figure 3: Edubuntu menu administrator.

Alphabetical overview

By default, Edubuntu applications are arranged alphabetically so that the students can see an example of alphabet sorting when they enter the system. Naturally, the applications addressing the same subject are grouped in the same folder, e.g. Art, Mathematics, Language and Learning, Science, etc. (see Figure 4).

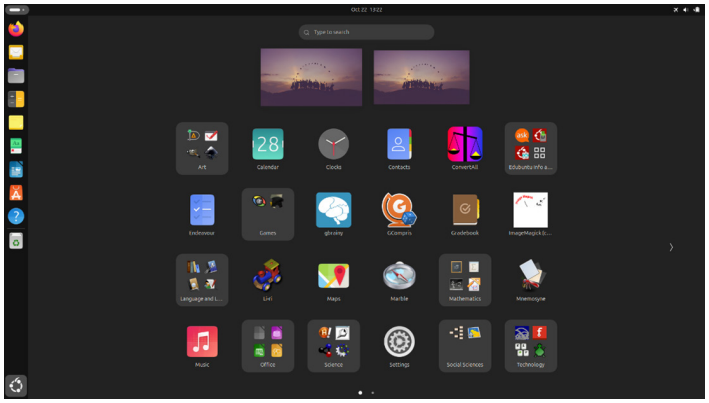

Figure 4: Edubuntu application overview.

Office applications

For general administrative tasks, most applications carry from the parent Ubuntu distribution. These applications make Edubuntu convenient for general use as well.

Web browser:	Similar to Ubuntu, Edubuntu is pre-installed with Firefox as a default Web browser.
Mail client:	Edubuntu selects Geary as a default mail client due to its simplicity and small footprint, which works well with older machines.
Office suite:	LibreOffice Writer, LibreOffice Base, LibreOffice Calc, LibreOffice Draw, LibreOffice Impress, and LibreOffice Math are available for administrative and productive tasks.
Image tool:	ImageMagic and Gimp handle simple and complex image manipulation tasks, respectively.
Others:	Calendar, Calculator, Dictionary, Endeavour, Notes, Maps, Music, and Weather are included in the default installation.

Educational applications will be explored in further detail in the next section.

Resource utilization

For OSS to be adequately accepted and integrated across educational sectors, it must be easy to access and use while requiring limited computational resources and power. It is the latter in which Edubuntu excels, thereby driving up the range of possibilities in which it could be properly installed and used. Importantly, Edubuntu consumes a small amount of CPU and memory resources. With the first author's machine, running it idly on Acer Aspire 3 (Intel Core i5-1135G7 @2.4GHz) notebook with clean installation takes only about 3% CPU load on two cores and only 1.3GB of RAM (see Figure 5). It is important to mention that Edubuntu can run on much older computers than our pilot test setting and it can also operate when offline.

As a result, it can play a huge role in communities which lack modern technological resources and Internet access. Stated another way, it can potentially play a significant role in society in addressing educational equity and access. As detailed throughout this manuscript, there is much potential educational power in this free and low-resource capacity learning platform.

Educational applications in Edubuntu 24.04

The educational applications in Edubuntu version 24.04, supporting 21st century skills, can be categorized into two primary groups: (1) Applications designed for teaching and learning specific subjects such as mathematics, science, social

Figure 5: CPU and memory consumption when running Edubuntu 24.04 idly.

sciences, and astronomy; and (2) Applications emphasizing particular skills like learning retention, critical thinking, logical reasoning, analytical skills, design proficiency, and typing speed. The applications are specifically designed for age groups ranging from early childhood to higher education. Notably, during system configuration, users have the option to select the student grade level, which results in the automatic installation of a specific set of applications tailored to one's individual educational needs.

Applications for specific subject areas

The system pre-organizes the applications in this category by grouping them together in the subject group folder, facilitating easy access for students. Each subject group is specifically aligned with the corresponding subject areas, which are not restricted to science and technology, but also extend to include the social sciences, languages, and arts. These subject groups are detailed below.

Mathematics

Several technology tools are included for mathematics in Edubuntu. For instance, Cantor serves as a front-end interface to manipulate mathematical expressions, supporting various kinds of back-end engines such as Julia, Kalgebra, Lua, etc. (KDE, n.d.). Cantor can handle advanced mathematical manipulation and visualization through programming. Moreover, Kalgebra handles algebraic expressions as inputs and plots them in the figure, whereas Kig generates geometrical plots from the menu bar. Kmplot is more general, accepting mathematical expressions and outputting advanced plot types like polar plots or differential plots. Kbruch is specifically designed for teaching fractions, and kids can also enjoy Tux Math games to improve their mathematical skills.

Science

Edubuntu offers three applications tailored to chemistry education. First, Kalzium provides information on the periodic table and atomic properties, allowing students to explore and enhance their learning retention. Second, Chemtool is equipped with a useful drawing tool for chemical molecules, capable of exporting drawings into various file formats. Third, Atomix, a chemistry learning game, promotes critical thinking by challenging students to construct given molecules by maneuvering atoms through a maze-like map.

For physics education, one application, named Step, functions as a motion simulation tool, educating students about the laws of physics through the connection of various objects and running simulations to observe their movements and interactions. Another tool is Light Speed, which simulates the characteristics of objects moving near the speed of light, fostering students' curiosity and enabling them to adjust speeds while observing physical changes.

Astronomy students will find delight in the Stellarium application, as it accurately depicts stellar constellations from any selected location worldwide. This tool ignites their imagination, allowing them to explore new positions by simply clicking anywhere on the sky screen while also providing accurate information about deep-space objects. Another valuable scientific tool is Rocs, which serves as a graph theory analysis tool. Rocs is instrumental in solving graph theory problems, such as the minimum vertex cover problem (Canabrava & Cord-Landwehr, 2015).

Social sciences

One such tool is 'Gramps', aptly named for its function as a genealogy tool, enabling users to display their family tree in various formats such as pedigree, fan chart, or descendant fan. Exploring one's own family history can contribute to the development of information literacy, a vital skill in the 21st century, as genealogy research necessitates the ability to locate, evaluate, and utilize information from diverse sources (Sheppard, 2012). Another tool, KGeography, is designed to assess students on their knowledge of city locations within a selected country through interactive map quizzes.

Marble virtual globe

With Marble, the virtual globe and world atlas application, students can explore diverse aspects of Earth, Geography, and History. The Atlas view offers the topography of the Earth, while clicking on city markers reveals concise details about each location. Importantly, adopting a learner-centered approach and fostering students' learning autonomy, Marble provides numerous options for map views to be selected and explored. For example, users can examine average rainfall patterns using the precipitation map view or analyze temperature variations through the temperature map view. Moreover, users have the capability to journey back in time and observe Earth's historical changes, such as those dating back to 1689, utilizing the historical map view. Marble also offers a Moon map feature for those who are interested in celestial exploration.

Language and learning

Learners can explore tools like Scribus, a professional desktop publishing tool ideal for creating books, brochures, and cards. A few e-book applications are included in this folder, including Calibre, an e-book management tool, that not only facilitates format conversion but also enables users to fetch e-books from the Internet. For those who want active and engaging learning, KLettres, Kanagram, KHangMan, and ulcc cater well for their needs. These tools not only enhance language learning experiences but also provide ample opportunities for active and engaging learning.

Arts

Gimp and Inkscape are commonly used for image manipulation and graphics design for professionals. Also in Edubuntu is LibreCAD, a specialized CAD/CAM design and drawing application. Dia is an application for drawing a flowchart or a diagram. In addition, kids can enjoy drawing on TuxPaint. These tools foster creative expression as well as enhance visualization skills that are increasingly valued and promoted in the innovation-driven, knowledge-based economies of the twenty-first century.

Applications emphasizing particular skills

There are numerous pre-package applications offering students the opportunity to enhance critical thinking, logical reasoning, and analytical skills. By adopting a gamified approach, learners are engaged in a diverse range of idea generation and creative endeavors. These applications help facilitate the process of analyzing and refining new ideas.

gBrainy

Games and challenges can elevate learner motivation by arousing curiosity and intrigue. One example is gBrainy, which is a brain teaser application with four types of exercises: (1) logic puzzles, (2) mental calculations, (3) memory trainers, and (4) verbal analogies. The application keeps historical scores for comparison among exercises.

gCompris

GCompris is designed to help children engage in analytical reasoning and develop an understanding of relationships between numbers while learning basic arithmetic functions. It offers more than 100 activities inside to enhance children's skills in computing, reading, arithmetic, science, geography, gaming, and other areas (KDE, 2024).

Technology

Fritzing is a circuit design tool that a teacher may utilize to teach electronics. The tool produces realistic artwork of a printed circuit board (PCB). During the design process, students can choose from various electrical components (e.g., resistor, transistor, integrated circuit, switch, LED) and board components from vendors such as Arduino, Spark, and Intel, and place them on the breadboard. Students can manually connect the components or request an autoroute to automatically wire the connections. Once the PCB design is completed, the schematic drawing can be exported for creating authentic prototypes. Additionally, the real PCB can be ordered directly from the application and manufactured by Fritzing Fab in Germany.

Three applications are included for teaching programming. One is KTurtle, which teaches programming basics by controlling the movement of a turtle through a program script. Basic256 is an IDE to teach middle to high-school students with the BASIC language. It has output windows in both text and graphics. The other is Laby, which teaches simple C, C++, or Perl languages by controlling the movement of an ant robot to find an exit from a labyrinth.

Elementary students are able to enjoy a typing game on Tux Typing. A mature learner may prefer a more formal practice typing tutor with Klavaro, providing correct typing instruction and improving typing speed.

Games

The gamification of learning is increasingly relied upon to engage learners. Edubuntu includes games suitable for young learners. Jigzo offers simple jigsaw puzzles, while Blinken is an educational game designed to help learners promote their cognitive abilities and pattern recognition skills. The game presents a sequence of colors and asks the player to memorize the order of flashing lights on the screen. As the levels progress, the sequences become longer and more challenging. In effect, Ri-li is a fun arcade-style game where players control a train's movement with precise input. Along the way, Ri-li helps learners improve their coordination between screen visuals and hand movements.

Mnemosyne

Flash cards are a widely used memory technique in learning. This application is a flash card study program designed to help learners remember new vocabulary or associations with images. It includes a flash card browser for managing flash cards, allowing users to schedule their learning sessions and view statistics, such as retention scores.

Edubuntu for teaching and learning

Edubuntu has the potential to be utilized in instructional design. The following section presents two examples of how Edubuntu can provide teachers with valuable tools to educate their students. Possible applications of Edubuntu in higher education are also discussed.

Teaching computational thinking with Edubuntu

Computational thinking (CT) has received enormous attention from educators and policymakers worldwide, aiming to integrate it effectively into early-level curricula (Kotsopolous et al., 2017). CT, regarded as the third fundamental element in scientific research alongside theory and experimentation (Cansu & Cansu, 2019), notably influences problem-solving methods (Oden & Ghattas, 2014). Essentially, CT explores merging human thought with computer technical capabilities to enhance problem-solving processes. Recent progress in computer science education includes the use of coding, graphical interfaces, hybrid models, and unplugged activities to foster K-12 learners' computational thinking skills (Taub et al., 2012; Voogt et al., 2015; Weintrop et al., 2016).

Scholars have examined the process of teaching computational thinking and synthesized it into a framework of pedagogical experiences for CT including unplugged, tinkering, making, and remixing (Kotsopolous et al., 2017). Unplugged experiences typically involve no technology, whereas the rest require the use of a computer device. Tinkering refers to making changes to existing examples to learn the effect of such changes. Essentially, tinkering entails trial and error approaches to continually refine or improve the process or product. In contrast, making refers to creating a new object or artifact from scratch. The process of making engages the learner in designing or generating products. Finally, remixing refers to the ability to reuse other objects as a component in solving a computational problem.

In addition to the teaching of CT, an article "Exploring Computational Thinking," from the Google Research team (Kao, 2010) defines the cognitive thought process of CT: decomposition, pattern recognition, abstraction, and algorithms. The process starts with a problem or task, which the student decomposes into several subproblems. Once recognizing repeated patterns or sequences, the student comes up with a reusable function and/or looping of the steps. Abstraction transforms the problems into commands and variables. Then, algorithms can be designed to solve each specific subproblem.

The pedagogical framework for CT provides guidance for teachers in creating effective learning activities (Kotsopoulos et al., 2017). Additionally, Google's cognitive thought process offers teachers a means to evaluate students' understanding and proficiency in CT (Kao, 2010). Notably, suggested Edubuntu applications may combine pedagogical roles and cognitive thought processes in designing the learning instruction, as summarized in Table 2.

To implement the Edubuntu application, GCompris is a great option for introducing young children to the basic programming concepts. For example, in the Simplified Tower of Hanoi game, children transfer stacks from one tower to another, and, in the process, they learn to create systematic and logical steps that process inputs and generate outputs. Such a procedure fosters algorithmic thinking through

Table 2: Edubuntu applications with their pedagogical roles and thought processes.

Application	Pedagogical Roles (Kotsopolous et al., 2017)	Thought Processes (Kao, 2010)
GCompris (Simplified Tower of Hanoi)	Tinkering, Making	Algorithms
GCompris (A sliding-block puzzle game)	Tinkering	Algorithms
GCompris (Path encoding /Path decoding)	Tinkering	Abstraction
Ri-li	Tinkering	Decomposition
GCompris (Programming maze)	Making	Algorithms, Decomposition
KTurtle	Tinkering, Making, Remixing	Pattern recognition, Abstraction, Algorithms
Laby	Tinkering, Making, Remixing	Decomposition, Pattern recognition, Abstraction, Algorithms

trial and error as they solve the problem. Another game, Programming maze, prompts children to guide a penguin to its destination by organizing a simple sequence of actions (see Figure 6). This activity teaches them to select actions and create foundational sequences, nurturing their algorithmic thought process.

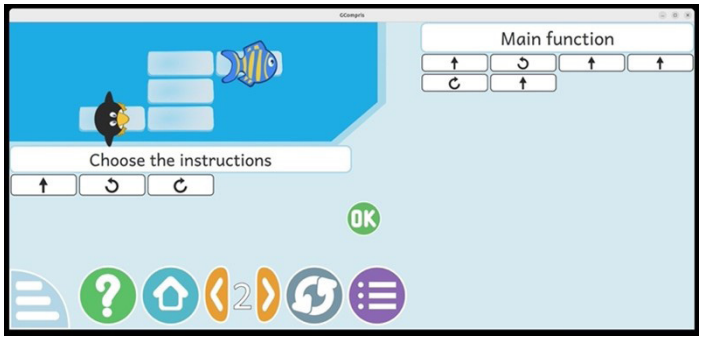

Figure 6: GCompris (Programming maze).

KTurtle is a valuable tool for teaching CT skills. Through scenario-based teaching, students engage in activities where they replicate movement patterns through coding scripts. Beginners can learn and experiment with TurtleScript examples provided. The application enables students to modify code and observe changes in the turtle's movements on the screen, fostering exploration and experimentation. Figure 7 illustrates how KTurtle utilizes TurtleScript for programming. The application also provides ready-made code for the students to reuse in their assignments. For example, if students want to set the canvas size to be 200x200 pixels and the canvas color to black, they can issue these commands:

```
canvassize 200,200
canvascolor 0,0,0
```

In the application, students begin creating their own code by learning concepts such as variable assignment, mathematical operations, Boolean logic, and comparing operators. For example, $n=1 and $answer = 10 > 3. Students can also attempt to remix the code by using these commands to move and control the turtle.

```
forward 20
turnright 30
```

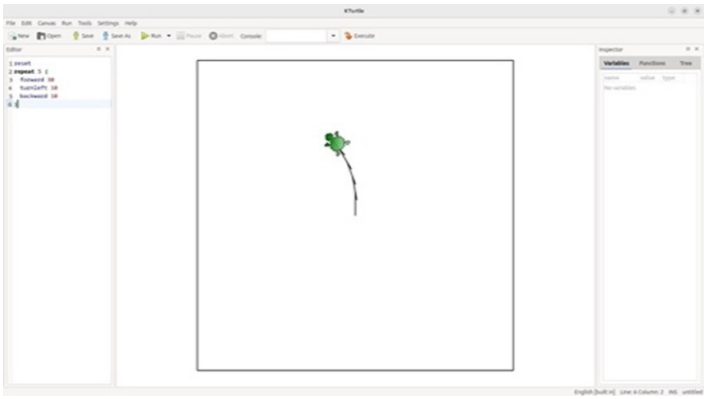

Figure 7: KTurtle.

Finally, students learn how the pattern of coding was created using the conditional expression and loop expression based on the provided code.

```
If ($n == 1) {
        print "hello 1"
} else {
        print "hello not 1"
}
repeat 5 {
        forward 3
        turnleft 1
        backward 2
}
```

For a more enthusiastic learner, one may practice programming exercises in Laby, in which the learner will write or tinker a code in C, C++, or Perl. The objective of the application is to move an ant robot to the exit while avoiding an obstacle. The learner will become familiar with the syntax of the programming language, attempt to debug a program, and observe the resulting run. An example of C code in Laby is shown below:

```
#include "robot.h"
void ant()
{
 right();
 forward();
 take();
 left();
 forward();
 drop();
 escape();
}
```

The C commands in the ant() function are taken from the robot.h library. They are written to serve tasks specifically in Laby.

Teaching geography with Edubuntu

Another prominent subject that Edubuntu can leverage is Geography. According to (Bednarz et al., 2013), geography is viewed as an essential skill/knowledge in the global economy and is mandatory in K-12 education in the United States. Geography is literally and figuratively a rather vast subject

area. It is not merely the factual information of specific places and locations. Geography addresses questions of "Where?" and "Why there?" (Roberts, 2003). It also involves the human, event, and time dimensions that interrelate with the locations. In Geography for Life, geography skills consist of: (1) asking geographic questions, (2) acquiring and (3) organizing geographic information, and (4) analyzing, and (5) answering geographic questions (Heffron & Downs, 2012).

With GCompris, young learners at primary school level attempt to locate the country and region in "Locate the countries" and "Locate the region" activities by matching the given shapes to the location on the map (see Figure 8). Moreover, young people can learn about important landmarks from the "Explore monuments" activity.

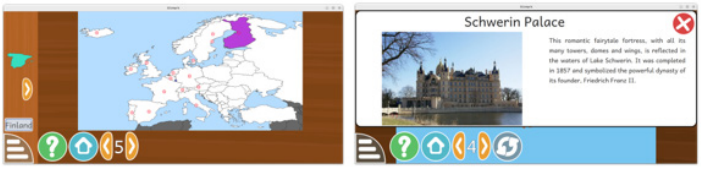

Figure 8: Locate the region (left), explore monuments (right).

A similar place-matching activity is present in KGeography, functioning at a more difficult level due to no boundary clues between the regions (see Figure 9). Knowledge acquisition of capitals and flags of provinces are included in the form of questions and answers.

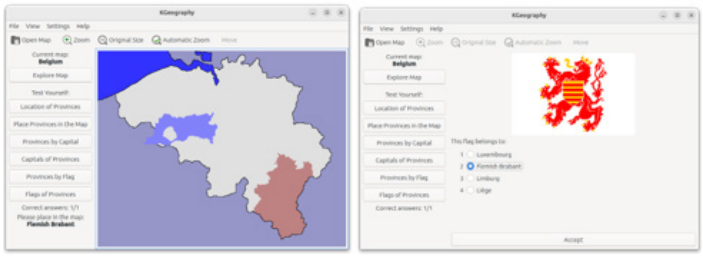

Figure 9: Region placement activity in KGeography (left), questions about flags (right).

A more advanced geography lesson can be conducted with Marble. Although Marble is not a Geographic Information System (GIS) tool, it can easily be used for inquiry-based learning. For example, the teacher may ask a student, "Where is Kota Kinabalu?" and "What is the distance between Bangkok and Singapore?" The student needs to query the location from the search box and then practice measuring the distance on the map (see Figure 10). Marble can also perform area measurements from a polygon created by measurement points. More sophisticated learners may create new maps by themselves that can be viewed on Marble. With some guidance from the teacher, students are able to develop their spatial thinking and can become active learners using the aforementioned geography skills.

Figure 10: Inquiry-based learning on Marble.

Edubuntu for higher education

Pre-service teachers, who enroll in an undergraduate Education program, will find Edubuntu particularly beneficial due to its comprehensive collection of software for K-12 and elementary schools. They can apply theoretical knowledge in teaching pedagogy and demonstrate it on the Edubuntu platform in various subjects as discussed earlier. Before employing it in a real classroom, pre-service teachers need to be competent in this technology-enhanced learning tool. Fortunately, many of the skills learned in Edubuntu are transferable to other general Unix OSs; for example, to the macOS which is well-known for all Apple notebook users.

The Edubuntu OS itself is worth an exploratory topic in the Operating Systems course in the Computer Science department. Since Edubuntu is a flavour of Linux, the students may be asked to create another flavour of Ubuntu which targets a different domain of interest, such as healthcare, entertainment, business and entrepreneurship, etc. By dissecting the system design of Edubuntu, the instructor can guide the development of different tools and features of new Ubuntu flavours such as the resource management, real-time scheduling, and interface design.

Analysis of Edubuntu under the instructional software design standard

The framework to design an educational OS is not found in the literature. Nevertheless, in this paper, we adopt the standards for instructional computing software design and development (Schaefermeyer, 1990) in order to analyze Edubuntu. Over three decades ago, Schaefermeyer (1990) identified the minimum characteristics that any educational software should include. Four key criteria are detailed below.

1. Design of learning activities

This standard refers to the matching of learning activities with the learning objectives. Also, the targeted audience should be specified. In Edubuntu, the learning objectives depend on each installed software tool, but they may not be clearly stated in the documentation as most instructions in the Help section only explain how to use such software. For the targeted audience, Edubuntu allows administrative setting options to select student levels, as mentioned in the Edubuntu 24.04 general specification section. After choosing the student level, the list of available applications is customized accordingly.

2. Identify the curriculum role used

There are three curriculum roles: (1) Adjunct, (2) Mainline, and (3) Management (Cohen, 1983). Adjunct role refers to a supplementary resource or instructional material to the regular curriculum. Mainline is a unit that is used as a complete subject, which may include tests and tutorials. Management relates to the learning records, data analysis, and measurement of outcomes. Edubuntu software primarily assumes an adjunct role as software developers for the various tools normally creates an application focusing on a particular knowledge or skill. In some applications, there may be a management feature in storing the learning progress or analyzing the learning outcome.

3. Identify mode of instruction to be employed which best achieves the objective

In the literature, there are five modes of instruction: (1) drill and practice, (2) tutorial, (3) games, (4) simulation, and (5) problem solving. Depending on each application, one or more modes of instruction can be facilitated. For example, GCompris includes all the modes of instruction while Light Speed! and Fritzing entail the simulation mode.

4. Other standard aspects that are found in Edubuntu include making the program menu driven, formatting instructional text for screen display, embedding graphics into content, using feedback appropriately, and employing random generation. These aspects and features are included in most Edubuntu applications.

Recommendations for educational OS design

From our scrutinization of Edubuntu and briefly scanning other educational OSs, we outline the following key consideration factors in designing an educational OS.

1. Light weight

Light weight means the OS consumes small CPU and memory resources. Due to the digital divide, the PCs or notebooks in remote schools do not generally have high specifications. They likely have relatively low-performance CPU and small amount of memory. To capture the greatest number of students, the OS should be light weight so it can run on an older machine.

2. Able to run offline

Again, due to the digital divide, the schools in remote area may have limited Internet access. The OS as well as its educational applications should be able to operate without the Internet. Once the OS is installed on the machine, it shall not require further online connection to run.

3. Packing with comprehensive educational applications

In general, the educational OS shall contain applications to support teaching and learning in several subjects and student levels. The purpose of each application may attempt to improve knowledge in a particular subject or skills in general such as creativity, problem-solving, logical thinking, etc.

4. Ease of installation or can run without installation

The OS installation shall be done by a person with basic computer knowledge in less than an hour. In the case of Edubuntu, the OS is packaged within a standard USB flash drive. The installation can be done easily on a machine with a standard USB port, without an Internet connection. It can even run by itself as well without installation in a trial mode.

5. Low-cost or free

Open-source Linux-based OS normally provides free license and usage. This fits well in schools or countries with low income. However, we should not interpret that the free OS will provide low-quality educational content. That normally depends on the application inside and many high-quality well-known educational software is free such as Scratch (Scratch Foundation, n.d.).

6. Attractive and colorful user interface

As with standard educational software, colorful icons and user interface attract students and learners to be more engaged in the learning content. The OS may include several themes that customize visual style.

7. Customizable according to the student level or interest

For example, the menu and list of software available can be customized according to the student level or interest. More granularity of the selected options enables more personalized learning. Ease of customization also encourages wider adoption of the OS.

8. Able to update or install a new application as needed

As knowledge and skills are constantly changing, new applications addressing those needs shall be easily downloaded from the Internet. A team of developers is required to constantly take care of the OS maintenance and software repository update, as well as new major and minor releases.

Conclusions

Various forms of media have demonstrated their positive impact on students' learning. However, each of these forms of learning media has distinct qualities that, when effectively

integrated, can enhance students' comprehension and skill development (Kozma, 1994). The lessons learned from the OLPC initiatives have emphasized that having a technology tool or device alone is insufficient. It is also crucial to incorporate pedagogical aspects in order to effectively facilitate students' learning and harness the potential of the tool in alignment with their educational objectives.

Without a doubt, the skills and competencies needed to successfully function and innovate in this global world are dramatically expanding. At the same time, new pathways to educational resources and opportunities are needed; in particular, for those in the Global South (Gunawardena, 2014; Zhang et al., 2020). Before Internet access expands globally (*The Economist*, 2019), the world community desperately needs to find and test low-cost solutions. Open-source software platforms like Edubuntu are one such solution. Since Edubuntu can be used in an offline mode, there are fewer technical issues and access concerns, making it an ideal solution for regions of the world with limited internet access and low-capacity hardware (Gunawardena, 2014; Zhang et al., 2020; Schotgues, 2022).

The tools and applications now available in Edubuntu are rapidly expanding and potentially transforming learning possibilities for learners ranging from the young to the more seasoned and from those living in the Global South to the Global North. The time is ripe to rapidly install and revisit research on this OSS platform, and, in the process, stretch human learning for many into new and uncharted territories where researchers can document novel uses of this learning platform as well as monitor and analyze significant areas of high impact and predict and promote pristine regions waiting to be explored and mined. The overhaul and rebirth of Edubuntu should be welcomed and embraced by educators, shared and utilized with learners, and documented and reported by researchers.

References

Ames, M. (2021, October 27). Laptops alone can't bridge the digital divide. *MIT Technology Review*. https://www.technologyreview.com/2021/10/27/1037173/laptop-per-child-digital-divide

Barbareev, K. (2015, September 24-26). Between traditions and innovations in the system of training of teachers [Paper Presentation]. *17th International BASOPED Conference - Traditions and innovations in the education of the Balkan countries,* Varna, Bulgaria.

Baru, M., Khan, F., & Iyer, S. (2009). Computer Masti: Use of FOSS for teaching computers in Indian primary schools. *National Conference on Open Source Software, Mumbai, India*. cm-ncoss09.pdf

Bednarz, S. W., Heffron, S., & Huynh, N. T. (2013). A road map for 21st century geography education. Geography education research. *Recommendations and Guidelines for Research in Geography Education*.

Butler, S., Gamalielsson, J., Lundell, B., Brax, C., Mattsson, A., Gustavsson, T., Feist, J., Kvarnström, B., & Lönroth, E. (2022). Considerations and challenges for the adoption of open source components in software-intensive businesses. *Journal of Systems and Software, 186*, 111152. https://doi.org/10.1016/j.jss.2021.111152

Canabrava, T., & Cord-Landwehr, A. (2015). *The Rocs handbook*. KDE documentation. https://docs.kde.org/stable5/en/rocs/rocs/index.html

Canonical Ltd. (2024). *Ubuntu flavours*. https://ubuntu.com/desktop/flavours

Cansu, S. K., & Cansu, F. K. (2019). An overview of computational thinking. *International Journal of Computer Science Education in Schools, 3*(1), 17-30, https://doi.org/10.21585/ijcses.v3i1.53

Cohen, V. B. (1983). Criteria for the evaluation of microcomputer courseware. *Educational Technology, 23*(1), 9-14. http://www.jstor.org/stable/44423772

Edubuntu. (2024, October 15). In *Wikipedia*. https://en.wikipedia.org/wiki/Edubuntu

Eickmeyer, E. (2022, December). Announcing Edubuntu revival. *Ubuntu Community Discourse*. https://discourse.ubuntu.com/t/announcing-edubuntu-revival/32929

Eickmeyer, E. (2024, April 25). Edubuntu 24.04 LTS released. *Ubuntu Community Discourse*. https://discourse.ubuntu.com/t/edubuntu-24-04-lts-released/44455

Graber, S. (2016, March 21). *Edubuntu 16.04 and beyond*. Ubuntu Mailing Lists. https://lists.ubuntu.com/archives/ubuntu-devel/ 2016-March/039281.html

Gunawardena, C. (2014, March 19). MOOCs: Students in the global south are wary of a 'sage on the stage.' *The Guardian*. https://www.theguardian.com/education/2014/mar/19/cost-barrier-students-global-south

Hansen, P. B. (1973). *Operating system principles*. Prentice-Hall Inc., United States.

Heffron, S. G., & Downs, R. M. (2012). *Geography for life: National geography standards* (Second Edition). Washington, DC: National Council for Geographic Education.

Huang, X., Zou, D., Cheng, G., Chen, X., & Xie, H. (2023). Trends, research issues and applications of artificial intelligence in language education. *Educational Technology & Society, 26*(1), 112-131. https://doi.org/10.30191/ETS.202301_26(1).0009

Hwang, G.-J., Xie, H., Wah, B. W., & Gašević, D. (2020). Vision, challenges, roles and research issues of artificial intelligence in education. *Computers and Education: Artificial Intelligence, 1,* 100001. https://doi.org/10.1016/j.caeai.2020.100001

Kao (2010, October 25). Exploring computational thinking. *Google Research Blog*. https://research.google/blog/exploring-computational-thinking

KDE (2024, December 8). *GCompris*. https://www.gcompris.net/index-en.html

KDE (n.d.). *Cantor - KDE's frontend to mathematical applications*. https://cantor.kde.org

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education, 3*, 154-171. https://doi.org/10.1007/s40751-017-0031-2

Kozma, R. B. (1994). Will media influence learning? Reframing the debate. *Educational Technology Research and Development, 42*, 7-19. https://doi.org/10.1007/BF02299087

Oden, T., & Ghattas, O. (2014, January). Computational science: The "third pillar" of science. *TAMEST News*. https://tamest.org/news/newscomputational-science-the-third-pillar-of-science/

OLPC. (2005). *About OLPC*. https://laptop.org/aboutolpc/

OLPC. (2013, August 5). *Software components*. https://wiki.laptop.org/go/Software_components

Özüsağlam, E., & Poşpoş P. (2016). Comparison of open source softwares in mathematics education. *Konuralp Journal of Mathematics, 4*(1), 225-238.

Pan, G., & Bonk, C. J. (2007a). The emergence of open-source software in China. *International Review of Research in Open and Distance Learning, 8*(1). https://doi.org/10.19173/irrodl.v8i1.331

Pan, G., & Bonk, C. J. (2007b). The emergence of open-source software in North America. *International Review of Research in Open and Distance Learning, 8*(3). https://doi.org/10.19173/irrodl.v8i3.496

Roberts, M. (2003). *Learning through enquiry: Making sense of geography in the key stage 3 classroom*. Geographical Association.

Rudolph, J., Tan, S., & Tan, S. (2023). War of the chatbots: Bard, Bing Chat, ChatGPT, Ernie and beyond. The new AI gold rush and its impact on higher education. *Journal of Applied Learning and Teaching, 6*(1), 364-389. https://doi.org/10.37074/jalt.2023.6.1.23

Schaefermeyer, S. (1990). Standards for instruction computing software design and development. *Educational Technology, 30*(6), 9-15. Standards for instructional computing software design and development | Educational Technology

Schotgues, B. (2022). Sub-Saharan teachers' conditions and circumstances: A review. *Journal of Applied Learning and Teaching, 5*(2), 10-22. https://doi.org/10.37074/jalt.2022.5.S2.5

Scratch Foundation. (n.d.). *About Scratch*. https://scratch.mit.edu/about

Sheppard, S. (2012). *Visualizing climate change*. http://dx.doi.org/10.4324/9781849776882

Souter, D. (2021, January 13). Inside the digital society: Lessons from little laptops. *LSE Blog*. https://eprints.lse.ac.uk/108802/1/parenting4digitalfuture_2021_01_13_one_laptop_per.pdf

Talib, M. A., Einea, O., Nasir, Q., Mowakeh, M. F., & Eltawil, M. (2019). Enhancing computing studies in high schools: A systematic literature review & UAE case study. *Heliyon, 5*(2), e01235. https://doi.org/10.1016/j.heliyon.2019.e01235

Taub, T., Armoni, M., & Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions in Computing Education, 12*(2), 1–29. https://doi.org/10.1145/2160547.2160551

The Economist. (2019, June 8). The second half of humanity is joining the internet: They will change it, and it will change them. The Economist. https://www.economist.com/leaders/2019/06/08/the-second-half-of-humanity-is-joining-the-internet

Ubuntu Community. (2013, May 27). *Ubuntuschools*. Ubuntu Wiki. https://wiki.ubuntu.com/Education/UbuntuSchools

van Rooij, S. W. (2019). Adopting open-source software applications in U.S. higher education: A cross-disciplinary review of the literature. *Review of Educational Research, 79*(2), 682-701. https://doi.org/10.3102/0034654308325691

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015) Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715-728. https://doi.org/10.1007/s10639-015-9412-6

Weintrop, D., Beheshti, E., Horn, M., Kai, O., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

Zhang, K., Bonk, C. J., Reeves, T. C., & Reynolds, T. H. (Eds.). (2020). *MOOCs and open education in the global south: Challenges, successes, and opportunities*. Routledge. https://doi.org/10.4324/9780429398919