EDITORS
**GEORGE HURLBURT** U.S. Federal Service (Retired), USA; gfhurlburt@gmail.com
**SOREL REISMAN** California State University, USA; sreisman@computer.org

# 3E Model: How to Use OERs to Enhance Teaching/Learning

**Henry Chan**, The Hong Kong Polytechnic University

*This article presents a 3E (Enrich, Extend, and Elevate) model for the effective use of open educational resources (OERs) as well as 10 selected OERs for computer science to enhance teaching/learning.*

Traditionally, university teaching is based on textbooks. With the development of the Internet, many Internet resources, particularly open educational resources (OERs), are now available to replace or supplement textbooks. As defined by UNESCO, OERs are[1]

> "...teaching, learning and research materials in any medium, digital or otherwise, that reside in the public domain or have been released under an open license that permits no-cost access, use, adaptation and redistribution by others with no or limited restrictions."

Here, we use a broader definition, including online resources from the Internet. Apart from complementing traditional teaching, OERs can enhance student learning in new ways, especially in higher education.[2] To promote the use of OERs, there are two fundamental questions. First, how can teachers find OERs effectively? Second, how can OERs be used to enhance teaching/learning? To answer these questions, this article shares some ideas and experiences.

Let's start with the 3E[3] (Enrich, Extend, and Elevate) model, shown in Figure 1. There are three core elements.

1. *Enrich:* enriching conventional lectures with OERs (during or before lectures)
2. *Extend:* extending classroom learning by studying and sharing OERs

*Digital Object Identifier 10.1109/MC.2023.3243447*
*Date of current version: 5 April 2023*

3. *Elevate:* elevating student innovative/creative thinking by creating OERs.

Let's discuss these elements.

The first element is *Enrich*, which aims to enrich lectures with OERs. This seeks to complement traditional lecture slides, for example, by playing a short video clip (for example, 3–5 min) after presenting lecture slides. The video can introduce a topic or explain a concept in more detail (for example, showing an animation or a real-life demo). Note that it is better to use a short video and ask questions after playing it. For example, a few multiple-choice questions can be asked (for example, through an electronic form). Students can respond through their notebook computers or mobile phones (for example, by accessing through a QR code). Student answers can be reviewed immediately; if most students cannot answer a question correctly, the topic can be further explained.

Apart from videos, other OERs can be used during a lecture (for example, animation tools). For instance, to teach computer algorithms (for example, sorting) more effectively, various animations are available to show how the algorithms work. An example is the Algorithm Visualizer website (see the "Algorithm Visualizer" section). A teacher can show the codes and animations to explain how different sorting algorithms work. The codes can be changed to show students different effects as well. In general, lectures can also be enriched by asking students to study OERs before a class. This aims to stimulate their interest and help them be better prepared. For example, short introductory videos can be used for this purpose (see the "Selected OERs for Computer Science" section).

> Apart from complementing traditional teaching, OERs can enhance student learning in new ways, especially in higher education.

The second element, *Extend*, aims to extend classroom teaching, encouraging students to learn further and from each other outside the classroom. Of course, teachers can provide OERs for students to study after a class. While traditionally, students can read textbooks after lectures, there are now many more multimedia materials or OERs from the Internet to complement classroom learning. Another good way to encourage students to share OERs is through a discussion forum (that is, encourage students to discover related OERs from the Internet). Indeed, most learning management systems used by universities provide discussion forums. Some participation marks (for example, 3–5% of the overall grade) might be awarded to encourage sharing.

Furthermore, posting OERs should be supplemented by student comments or a brief learning reflection, for example, why the OERs are helpful. This seeks to enhance their critical thinking and analysis skills as well. For instance, a student may need help understanding specific topics after a lecture. They may find OERs on the Internet to facilitate their learning and share them through a discussion forum. While a teacher can find OERs for teaching/learning, it is beneficial to discover OERs through students and from their perspectives. Of course, if suitable, the suggested OERs can also be used for future teaching. Besides sharing OERs through discussion forums, sharing them through social networks can be a good alternative. Students like to share photos and videos this way. Why not encourage them to share OERs as well?

The third element is *Elevate*, which intends to move student learning to a new level or dimension. We call this method *flipped learning*. Students are guided to create or make OERs to teach others (that is, like a teacher). For instance, after learning a computing topic, students can create a short video lecture to "teach" others in their own ways. In particular, students are encouraged to think innovatively and creatively. To motivate students to create OERs, student contests can be organized. Indeed, IEEE COMPSAC has organized student OER contests. One of

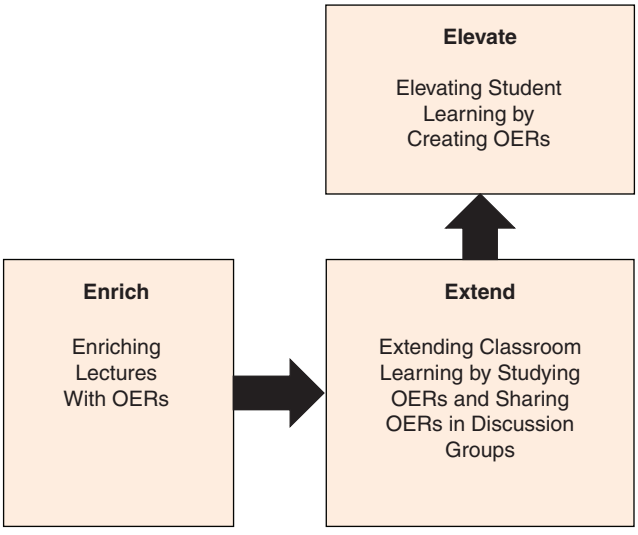


**FIGURE 1.** The 3E model.

the winning students created a video to use card magic to teach a computing algorithm creatively, then explained the concept using a computing concept, and finally, showed a Python program to demonstrate the algorithm.

Furthermore, internal OER competitions[4] can also be organized to enhance student learning. For example, we have organized a two-part OER student competition: 1) suggesting OERs and 2) creating OERs. For the first part, a student chooses a computing topic, finds a good OER from the Internet, and writes a learning reflection. For the second part, a student creates an OER (for example, a short video) to try teaching others (that is, using the flipped learning approach). An interdisciplinary judging panel was formed to select the winners. For suggesting OERs, students were evaluated based on their learning reflections according to the following criteria: critical thinking (30%), effective communications (20%), learning impact (20%), innovation (15%), and usefulness of the suggested OER (15%). For creating OERs, the following evaluation criteria were used: critical thinking (40%), effective communications (30%), innovation (15%), and usefulness of the created OER (15%).

## SELECTED OERs FOR COMPUTER SCIENCE

Now, you may ask—how can applicable computer science OERs be found? One source is the Multimedia Educational Resource for Learning and Online Teaching (MERLOT) Computer Science Community Portal (https://www.merlot.org/merlot/ComputerScience.htm). A distinctive feature of MERLOT is that there are both user ratings and peer reviews. Like research journals, OERs are managed by an editorial board but for teaching/learning purposes. Based on the user statistics from MERLOT and other information,[3,4] here are 10 alphabetically listed computer science OERs, which may be helpful for teaching/learning.

### Algorithm Visualizer

The first OER is the Algorithm Visualizer (https://www.merlot.org/merlot/viewMaterial.htm?id=773403099). Algorithms (for example, sorting algorithms) are essential in computer science. With a helpful visualization tool or interface, students can learn different algorithms through this website. They can study the essential operation of an algorithm by running the code and viewing an animation. It can be a practical demo for a data structure or algorithm course. As mentioned previously, this OER is well suited for enriching a lecture (that is, related to the "Enrich" element of the 3E model).

### Android Developer Fundamentals

The second OER is Android Developer Fundamentals (https://www.merlot.org/merlot/viewMaterial.htm?id=1378815). With the advent of smartphones, many students like to learn about app development. With well-structured lessons and hands-on exercises, this online course facilitates learning about the basics of Android app development. It can be used with mobile computing courses, particularly for extending classroom learning (that is, related to the "Extend" element of the 3E model).

### Articles on Computer Science Subjects

The third OER is the Articles on Computer Science Subjects website (https://www.merlot.org/merlot/viewMaterial.htm?id=773403097). Junior computer science students must have a strong foundation in computing, especially data structures, and algorithms. Using an integrated learning approach, this website provides a wide range of self-learning materials/resources, such as videos, exercises, and programs. For example, after attending a lecture, students can be directed to learn more about these topics through this website (that is, related to the "Extend" element of the 3E model).

### C Programming Exercises, Practice, Solution

The fourth OER is the C programming Exercises, Practice, Solution website (https://www.merlot.org/merlot/viewMaterial.htm?id=773403098). C is a fundamental and general programming language. This website not only facilitates students to learn how to write C programs but also provides many exercises, including programming solutions for practicing programming. An easy-to-use web interface is provided for writing and running C programs. For example, in relation to the "Extend" element of the 3E model, students can be guided to practice programming using this website.

> Posting OERs should be supplemented by student comments or a brief learning reflection, for example, why the OERs are helpful.

### Computer Science Videos

The fifth OER is the Computer Science videos website (https://www.merlot.org/merlot/viewMaterial.htm?id=773403094). To stimulate student interest in computer science, this channel provides over 40 videos covering different computer science topics. These videos can complement computing lectures and student self-study (for example, before a class). For example, the videos are particularly suitable for enriching a related lecture (that is, before a class, students can watch some videos to stimulate their learning interest).

### E-books Directory

The sixth OER is the E-books Directory website (https://www.merlot.org/

merlot/viewMaterial.htm?id=862157). There are now many open textbooks (that is, OERs). This website provides a directory of free e-books covering multiple subjects, including computer science. Students can find and access e-books through a single interface. This is related to the "Extend" element of the 3E model in two ways. First, students can read further references after a class. Second, students can also recommend or share their favorite e-books (that is, sharing OERs).

### SQL Tutorial

The seventh OER is the SQL Tutorial website (https://www.merlot.org/merlot/viewMaterial.htm?id=83289). Structured Query Language (SQL) is an important query language for database management. This website provides comprehensive materials, including examples and exercises for learning SQL. Students can run SQL commands through a web interface and see the results quickly. Students can learn from a wide range of examples. It can be used in conjunction with a database course. For example, a teacher can enrich a database lecture by showing different SQL statements through this website.

### MIT App Inventor

The eighth OER is the MIT App Inventor website (https://www.merlot.org/merlot/viewMaterial.htm?id=763912). Based on graphical programming, MIT App Inventor provides a user-friendly web-based interface for developing mobile apps. Apps can be created easily by putting programming blocks together (that is, similar to LEGO blocks). It is a good tool for first-year students to learn about computational thinking (for example, before learning other programming languages). For example, in relation to the "Elevate" element of the 3E model, it can be used to elevate student learning by creating interesting and innovative apps for teaching computational thinking.

> While a teacher can find OERs for teaching/learning, it is beneficial to discover OERs through students and from their perspectives.

### Orange

The ninth OER is the Orange application (https://www.merlot.org/merlot/viewMaterial.htm?id=773310901). This application provides a graphical programming tool for analyzing data and conducting data mining and machine learning. It can be used in an introductory course in machine learning. For example, students can set up different machine learning models to train data and predict results. Like MIT App Inventor, it can elevate student learning in another aspect (that is, data-oriented learning).

### Python Tutor

The 10th OER is the Python Tutor website (https://www.merlot.org/merlot/viewMaterial.htm?id=1174990). Python is one of the fundamental programming languages for computer science students (primarily first-year students). This website teaches students about Python and other programming languages, including JavaScript, C, C++, and Java. Through the web interface, students can write and run programs. It is particularly useful for supporting programming courses like the OER mentioned previously related to C programming.

In conclusion, the 3E model (that is, enriching lectures with OERs, extending student learning by sharing OERs, and elevating student learning by creating OERs) should provide a basic framework for using OERs to complement traditional teaching/learning. Indeed, there are many good OERs available on the Internet. Apart from the good examples presented previously, you can find more OERs from MERLOT. These OERs cover not only different topics but also use different approaches. You are also encouraged to find more OERs on the Internet. C

### REFERENCES


1. E. Tovar and N. Piedra, "Guest editorial: Open educational resources in engineering education: Various perspectives opening the education of engineers," *IEEE Trans. Educ.*, vol. 57, no. 4, pp. 213–219, Nov. 2014, doi: 10.1109/TE.2014.2359257.
2. T. Wang and D. Towey, "Open educational resource (OER) adoption in higher education: Challenges and strategies," in *Proc. IEEE 6th Int. Conf. Teaching, Assessment, Learn. Eng. (TALE)*, Hong Kong, China, 2017, pp. 317–319, doi: 10.1109/TALE.2017.8252355.
3. H. C. B. Chan, Y. H. Ho, E. Tovar, and S. Reisman, "Enhancing the learning of computing/IT students with open educational resources," in *Proc. IEEE 44th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, 2020, pp. 113–122, doi: 10.1109/COMPSAC48688.2020.00024.
4. H. C. B. Chan, I. M. Kemp, W. C. L. Leung, E. Tovar, and S. Reisman, "Enhancing student learning through an open educational resource competition," in *Proc. IEEE 45th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, 2021, pp. 127–132, doi: 10.1109/COMPSAC51774.2021.00029.


**HENRY CHAN** is an associate professor and associate head of the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. Contact him at cshchan@comp.polyu.edu.hk.