EFCEE'99

THE ASTEP DISTRIBUTED FRAMEWORK FOR TEACHING AND LEARNING

J. M. FERREIRA and M. A. SANTIAGO [1]
L. MACKINNON and P. MCANDREW [2]
OYSTEIN RA and T. STROM [3]

1) University of Porto (FEUP / DEEC) – Rua dos Bragas – 4099 Porto Codex - PORTUGAL
2) Heriot-Watt University – Riccarton – Edinburgh EH14 4AS – SCOTLAND
3) Buskerud College of Engineering – P. O. Box 251 - Frogs vei – 3601 Kongsberg – NORWAY

Abstract - ASTEP is an Educational Multimedia project supported by the European Commission and promoted by a consortium that includes companies and educational institutions from five European countries. The ASTEP mission is twofold: i) to deliver high-quality multimedia training materials to trainees in process-based industries, within a collaborative, task-centric, communication-based environment; ii) to use that environment to support industrial and academic trainers in the authoring and redevelopment of those training materials. To achieve these objectives, the ASTEP project has under development a framework with four main components: i) a model for learning; ii) an instructional design approach; iii) the support technology and iv) the framework technology itself.

INTRODUCTION

Internet-based course delivery became increasingly important in the last years and is currently competing for its share in the education and training market [1-7]. As part of the on-going work in this field in Europe [8], the ASTEP (Advanced Software for Teaching and Evaluation of Processes) framework is under development in a two-year project that started in February of 1998 [9]. This overall framework is task-centric, in the sense of assuming that every action taken by the users are in response to learning tasks that make explicit the course knowledge and identify what progress took place toward an identified learning outcome. In order to accommodate different training backgrounds, all courseware comprises a multi-layer information structure, where the level of detail depends on the users' previous expertise level.

This paper will start by presenting the ASTEP learning model, with emphasis on the courseware and its task-centric nature, followed by the instructional design support facilities that are available to the tutors. All pedagogical aspects were of primordial importance on the definition of the overall framework, which relies on a technological infrastructure able to accommodate a wide range of tools. The description of the underlying technology resources is then presented, followed by a concluding section and a list of relevant bibliographical references.

THE ASTEP LEARNING MODEL

The ASTEP learning model describes the educational philosophy adopted within the project and is based on a three-stage process with the following steps:

- **Conceptualisation**, where learning takes place through an interaction between the learners' pre-existing framework of understanding and a new exposition. Learning materials are made available to the students in computer-based form, which are described in the ASTEP framework as *primary courseware*.

- **Construction** refers to the application and testing of new concepts, in the performance of meaningful tasks. *Secondary courseware* resources are available at this stage to help the learner perform the learning tasks.
- **Dialogue**, the third stage, covers the process of creation and testing of new conceptualisation during interaction with the tutors and with other learners. *Tertiary courseware* is associated with this stage and consists of the materials produced by the learners themselves during the execution of their learning tasks.

The three-stage ASTEP learning model was based on the work of Professor Terry Mayes [10] and is illustrated in figure 1, where the resources required for the management of the learning cycle are also shown. This model brings into evidence that the construction stage is task-driven. All the material presented to the user and all the actions that the user takes relative to that material is associated with learning tasks. The completion of these tasks will mean that the user has progressed towards an identified learning outcome. While being central to the environment from the perspective of the user, the tasks are not the governing concept of the framework. Rather, they are the outcome of the use of the framework to develop the learning environment for a particular domain or subset of that domain. Course knowledge, resources and technology are the guiding factors relative to any domain learning development, which will determine the tasks to be undertaken by the users.

INSTRUCTIONAL DESIGN SUPPORT

Notice that the identification of the nature and type of the tasks to be developed, together with the material from which to develop them and the resources and technology to be used in their development, is necessary but not sufficient to guarantee the generation of appropriate and useful tasks, leading the trainees to the desired learning outcomes. There is a need to provide guidance and support for trainers in authoring material and developing new material using alternative media. The intention in developing the ASTEP framework is that this process should be consistent and repeatable, and that there should be no requirement for specialist computing or multimedia skills on the part of the trainers or the users of the framework.

The overall task construction process builds upon a task data model described in table 1. This data model could be represented by a meta-data set provided through XML, to support future tools that will aid task reuse and construction.

Once a task has been defined according to the data model described above, the task construction process can take place as shown in figure 2. The ASTEP framework includes tools to support the task construction process, which provide guidance on the structure and validation that can be applied.

With the objective of exemplifying the instructional design support tools available within the ASTEP framework, a case study describing the development of interactive digital circuit examples will now be presented in the following section.

ISVs: A CASE STUDY OF INSTRUCTIONAL DESIGN SUPPORT

ISVs (Interactive Student Views) are a simulation resource that a trainer can use to enable a "hands-on" activity, with the objective of helping the students to better understand the circuit behaviour under specified conditions (including faulty conditions). ISVs are seen by the student as shown in figure 3 for the simple example of a three-input combinational circuit.

The following elements are represented in each ISV:

1. The circuit, including its primary inputs and outputs, the basic logic gates — AND, OR, etc. — and their interconnections.
2. Hot-Spots (**HS**), indicated by a small circle over the respective circuit node, which in figure 3 are of the two following types:
   - HSs associated to the circuit primary inputs (**HS-I**) — these HSs have their visualisation window always open (showing the logic value present at the node; the initial logic value in a node is **0** by default). Clicking on the HS-I with the left button of the mouse toggles the logic value present in the primary input.
   - HSs associated to the output of combinational blocks (**HS-C**) — the default state of a HS-C visualisation window is open (in figure 3 all visualisation windows in the HS-Cs were previously closed). Clicking on the HS-C with the left button of the mouse toggles the state of the visualisation window, from open to close and vice-versa.

Two additional types of HSs are available to enable the representation of synchronous sequential circuits, namely for clock signals (**CLOCK**) and flip-flop outputs (**HS-S**).

The guidelines that were set up concerning the development of instructional design support for ISVs may be summarised as follows:

- The procedure generating the ISVs (for combinational or synchronous sequential circuits) must be systematic and as simple as possible.
- The instructor building the ISV shouldn't necessarily be skilled in JAVA programming or in HTML code details.
- The information required from the instructor should be provided in a small ASCII file for each ISV (the *ISV description file*), containing a brief circuit description and layout information.

The ISV description file built by the instructor (using a standard text editor) is used as input to an application (written in C++) that automatically generates an HTML applet tag. This HTML applet tag is then copied and pasted into the HTML page of the course module and will instruct the browser to call a JAVA applet that simulates the required combinational or sequential circuit (this HTML applet tag also contains the parameters to be passed to the JAVA digital circuits simulator, for each ISV). The availability of these two applications greatly simplifies the generation of ISVs, in accordance with the objectives presented earlier. The sequence of steps to generate an ISV may therefore be described as follows:

1. Using a standard text editor, build the ISV description file, which contains two sections: a description of the circuit and a description of the ISV layout.
2. Run the application that produces the HTML applet tag instructing the browser to call the JAVA applet for digital circuit simulation.
3. Copy and paste this HTML applet tag (containing the parameters describing the circuit to be simulated) into the proper place in the HTML page that will contain the ISV.

This sequence of steps is always the same, both for combinational and sequential circuits, under faulty or fault-free conditions.

To illustrate the process of generating an ISV using the HTML applet tag generator and the JAVA digital simulator, a simple sequential circuit is represented in figure 4. The ISV description file that must be provided by the instructor includes the two sections previously described (circuit description and layout description) and may in this case be presented as follows:

```
INPUT Q0
INPUT Q1
INPUT A
NOT NQ0 Q0
NOT NQ1 Q1
AND E Q0 Q1
AND F NQ0 NQ1
OR G E F
BUF D0 Q1
BUF D1 A
.SIZE 515 515
.IMAGE sequencial.gif
.INS 1
.OUTS 1
.FFD 2
.CLOCK 360 449
.PRINT D0 hss.1 125 349
.PRINT D1 hss.2 125 468
.PRINT A hsi.1  49 39
.PRINT G hsc.1  479 140
```

The HTML applet tag generator (using as input this ISV description file) produces the following output:

```
<APPLET code="simulator.class" width="515" height="515">
<PARAM name="image" value="images/Image36.gif">
<PARAM name="n_inputs" value="1">
<PARAM name="n_outputs" value="1">
<PARAM name="n_ffd" value="2">
<PARAM name=clock.x value=360>
<PARAM name=clock.y value=449>
<PARAM name=hss.1.x value=125>
<PARAM name=hss.1.y value=349>
<PARAM name=hss.1.v value="yes">
<PARAM name=hss.1.t value="0,0,0,0,1,1,1,1,D,D,D,D,/D,/D,/D,
/D,0,0,0,0,1,1,1,1,D,D,D,D,/D,/D,/D,/D,0,0,0,0,1,1,1,1,D,D,D,
D,/D,/D,/D,/D,0,0,0,0,1,1,1,1,D,D,D,D,/D,/D,/D,/D">
<PARAM name=hss.2.x value=125>
<PARAM name=hss.2.y value=468>
<PARAM name=hss.2.v value="yes">
<PARAM name=hss.2.t value="0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,/D,
/D,/D,/D,/D,/D,/D,/D,/D,/D,/D,/D,/D,/D,/D,/D">
<PARAM name=hsi.1.x value=49>
<PARAM name=hsi.1.y value=39>
<PARAM name=hsi.1.value value="0">
<PARAM name=hsc.1.x value=479>
<PARAM name=hsc.1.y value=140>
<PARAM name=hsc.1.v value="yes">
<PARAM name=hsc.1.t value="1,0,/D,D,0,1,D,/D,/D,D,1,0,D,/D,0,1,1,0,
/D,D,0,1,D,/D,/D,D,1,0,D,/D,0,1,1,0,/D,D,0,1,D,/D,/D,D,1,0,D,/D,0,1
,1,0,/D,D,0,1,D,/D,/D,D,1,0,D,/D,0,1">
</APPLET>
```

Looking at the code sequence above, we note that:

- The name of the applet class (in this case "simulator.class") is defined first and the applet size is allocated on the browser page using the width and height attributes (the values used are those referred for the image size).
- The Image parameter is then used to define the URL of the image to be displayed.

- The number of circuit inputs, circuit outputs and flip-flop devices is then presented.
- The definition of the HSs starts with the HS-clock, located in position (360,449). The two existing HS-Ss are located at (125,349) and (125,468), in both cases with an open visualisation window. The only HS-I existing in this case is located at (49,39), with initial value of **0**, and the only HS-C, which is also a circuit primary output, is located at (479,140), with an open visualisation window.

Notice that the functional description of each HS-C and HS-S considers an extended logic set (0, 1, D and /D), to enable the representation of single stuck-at faults.

TECHNOLOGY RESOURCES

Each module will operate within a Learning Environment that brings together the management of students, tasks, resources, assessment, and discussion support. Within the project there is experience with several environments, both custom and available from other suppliers, including WebCT, TopClass and Lotus Learning Space. The environment will be selected and configured to offer explicit support for the task and communication framework.

Communication support will be integrated and matched to the technology available. It is envisaged that a key component of this will be discussion support, and again various tools have been used, including HyperNews, FirstClass and WebBoard. Sychronous tools to support text based communication will also be incorporated, real-time communication with audio and video will form an aspect of the course and also explored for support, where appropriate technology is in use. Course components and possible technologies are shown in table 2.

The primary mode of delivery will be from a web-site dedicated to the provision of course material. This site will incorporate a learning environment that allows students to register and be monitored through their access to the system. This can be replicated through servers within company Intranets as well as available on the Internet. Opportunity for discussion can be provided by a web-supported computer conferencing system, such as WebBoard.

Use of discussion is an integral part of the framework and must be well supported. The discussion system is also designed to act as a focus for the learners' participation in a course and will provide routes through to help on the discussion lists as well as through email, telephone and synchronous links.

The task basis leads to a strong role for additional resources that can be catalogued on the web server and also actively sought by participants as they progress through early tasks. It is also important to make sure that resources are appropriate and it is foreseen that print and other material may be recommended to the learners.

The framework itself needs to be supported by appropriate technology. This will build upon the communication technology available, along with the support for other activities of the various actors in the learning process. Figure 1 can be redrawn to show the classes of activity within each section, leading to the representation shown in figure 5. A wide variety of resources are available to implement the ASTEP framework courseware at the various levels, giving the users the possibility of choosing the tools of their preference, provided that the requirements of each courseware component are met.

CONCLUSION

The famous American psychologist B. F. Skinner, whose work led in the 50s to the development of numerous "teaching machines" [11], is credited with a statement that could well be applied to describe the impact of Internet technologies in the education and training domain: *The real problem is not whether machines think but whether men do.* Keeping this statement in mind, the specification and development of the whole ASTEP framework was user-centred from the very beginning. It is therefore appropriate to conclude this paper with a general representation of the users' view of ASTEP, which may be represented as shown in figure 6 and is meant to be read from the inside out.

- At the core is the individual user with a computer providing local resources and material which can be worked on individually by that user.
- The next layer supports collaborative working with peers, in the form of discussion and news groups, project teams, and the reuse of problems, dialogues and solutions from previous groups.
- The third layer provides the interface to the various trainee support mechanisms, be they local to the user site or external, as in the case of a local college providing support for trainees within a process plant. These mechanisms can be both synchronous and asynchronous, individual and group, and could be specifically task-related or providing more general background and support.
- The final layer in the diagram illustrates the interface to the wider world outside of the immediate support environment. This is still likely to be constrained, in terms of the environment, to encompass domain experts, hardware and software suppliers, specialist news groups, or other sources of more detailed or specialist knowledge relative to the domain of interest.

Once the framework components are finalised and the first set of training materials is made available, the training effectiveness will be assessed by trial courses that are being organised among the consortium industry members.

REFERENCES

[1] B. Collis, *Tele-Learning in a Digital World – The Future of Distance Learning*, International Thomson Computer Press, 1996 (see also "On-Line and Distance Learning" on http://www.to.utwente.nl/ism/online96/campus.htm).

[2] Herb Bethoney, "Computer-Based Training on the Web," *PCWeek*, August 1997 (http://www.zdnet.com/pcweek/acrobat/1997/97pcweek/aug18mb.pdf).

[3] B. Landon, *On-Line Educational Delivery Applications: A Web Tool for Comparative Analysis* (http://www.ctt.bc.ca/landonline/).

[4] TeleEducation NB, *A Survey of New Media Development and Delivery Software for Internet-Based Learning*, 1997 (http://telecampus.edu/faculty/newmedia/).

[5] Robin Mason, *The Globalisation of Education*, 1997. (http://www-iet.open.ac.uk/pp/r.d.mason/GlobalEdu.html)

[6] W. D. Graziadei, S. Gallagher, R. Brown, J. Sasiadek, *Building Asynchronous & Synchronous Teaching-Learning Environments: Exploring a Course / Classroom Management System Solution.* (http://137.142.42.95/west/aslpaper.html)

[7] Ralph Ginsberg, Kenneth Foster, "The Wired Classroom," *IEEE Spectrum*, August 1998, pp. 44-51.

[8] *Multimedia Access to Education and Training in Europe – Memorandum of Understanding*, November 1998. (http://www2.echo.lu/telematics/education/en/news/mou.html)

[9] J. M. Ferreira, L. MacKinnon, M. Desmulliez e P. Foulk, "A Multimedia Telematics Network for On-the-Job Training, Tutoring and Assessment," *Proceedings of the International Conference on Engineering Education (ICEE'98)*, Rio de Janeiro, Brazil, August 1998.

[10] Terry Mayes, *Learning Technology and Groundhog Day*, in V. Strang & D. Slater (editors), *Hypermedia at Work: Practice and Theory in Higher Education*, Canterbury: University of Kent Press, 1995.

[11] B. F. Skinner, "The Science of Learning and the Art of Teaching," *Harvard Educational Review*, Vol. XXIV, No. 2, Spring 1954, pp. 86-97.

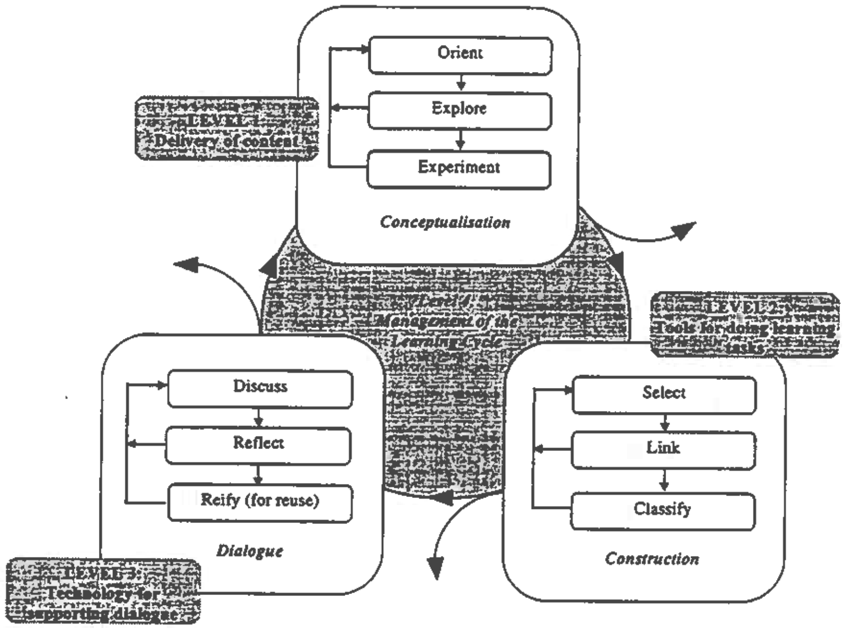

Figure 1: The ASTEP learning model.

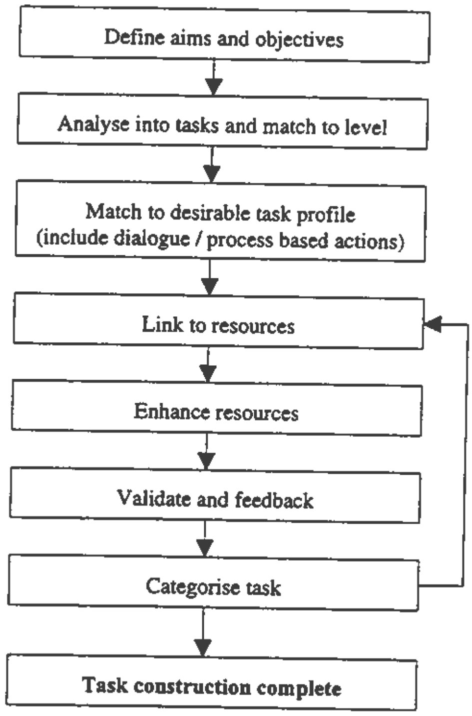

Figure 2: The task construction process.

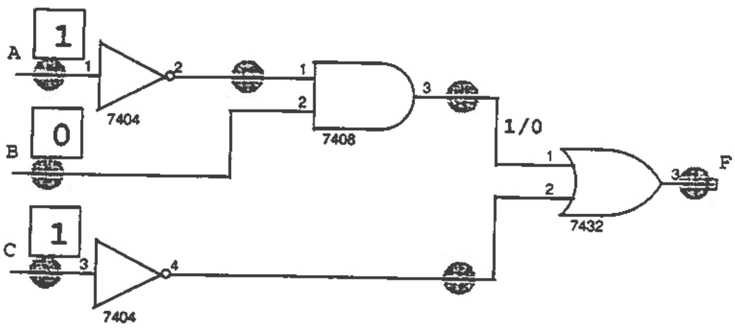

Figure 3: Student view of a simple combinational ISV.

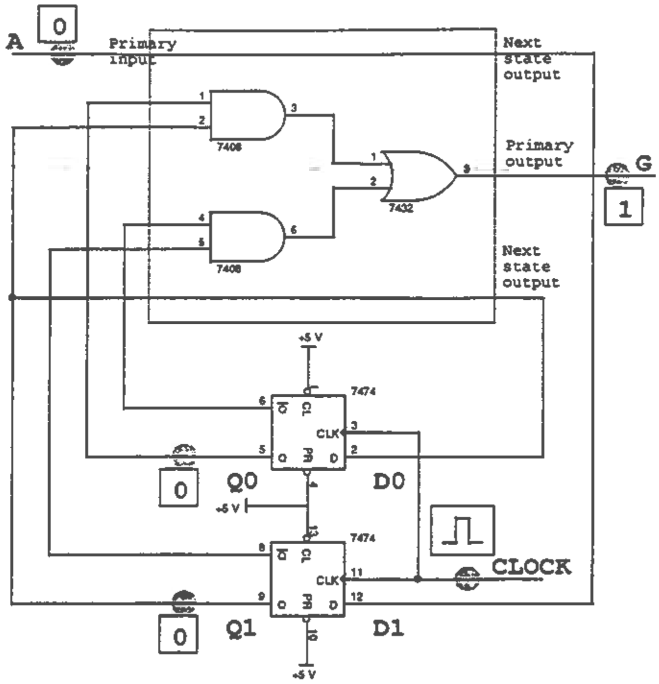

Figure 4: A simple sequential circuit.

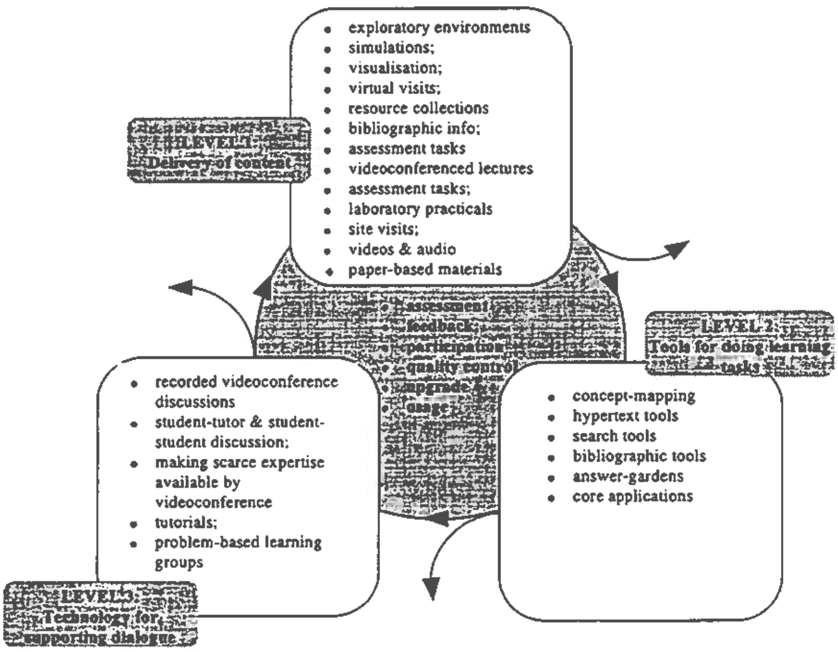

Figure 5: Courseware examples associated with each stage in the ASTEP learning model.

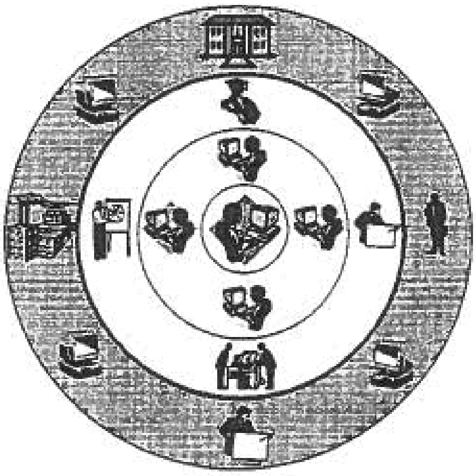

Figure 6: Users' view of the ASTEP framework.

Task model	Practical, analytical, specialist, applicative, discursive, analogous
Task category	Active, reflective, collaborative, dialogue
Task level	Advanced, detailed, overview
Hierarchy position	Parent task, dependent tasks
Sub tasks	Breakdown into parts
Resources required	Materials, tools
Schedule information	Timing, length
Discussion support	Lists, tutors, experts
Assessment	Options (maybe user dependent)
Aims	Match to learning objectives

Table 1: Tasks data model.

Component	**Possible technology**	**Actor**
Client	HTML browser – e.g. Netscape or Microsoft Internet Explorer	Learner
Learner management	WebCT	Learner/tutor
Task support	Customised support	Learner
Material	HTML incorporating other media (audio, video, presentations). Conversion from SGML.	Tutor
Discussion	WebBoard, WebCT, HyperNews, email	Learner/tutor
Simulations	Java applets, MultiVerse simulation toolkit	Learner/tutor
Collaborative tools	Custom whiteboard, Microsoft Netmeeting	Learner/tutor
Video conferencing	ISDN compatible (e.g. Intel Proshare)	Learner/tutor

Table 2: Possible technologies to implement the ASTEP framework.