AAEE 2012 CONFERENCE
Melbourne, Australia
www.aaee.com.au/conferences/2012/

Teaching Tools as Teaching Tools: Contextualised Authentic Learning Examples

Nickolas J.G. Falkner and Katrina Falkner
School of Computer Science, The University of Adelaide, Adelaide, SA, Australia, 5005
Corresponding Author Email: nickolas.falkner@adelaide.edu.au

BACKGROUND

Authentic learning examples are a well-established motivation and engagement tool for Software Engineering students. Numerous studies identify the use of industry case studies, software tools and current media stories as appropriate sources for authentic learning examples within the curricula. Assessment requirements dictate that we must carefully select real-world problems that have a known completion time or, when multiple problems are used across different groups, well-defined levels of difficulty. Where we can, and do, use project-type courses we can readily integrate industry-associated projects and other authentic problems, but this is still not of benefit to the majority of students. Contemporary teaching activities often require a large percentage of faculty to be involved in the development or selection of learning management systems, teaching tools and teaching support systems as part of their day-to-day work: the staff are immersed in an authentic design, selection and production environment that is, effectively, shielded from the students.

PURPOSE

In this paper we examine the use of software tools from the student learning environment as a source for contextualised learning examples, in terms of their ability to add authenticity to the learning experience, and assist academics in efficiently sourcing relevant learning examples.

DESIGN/METHOD

We present an experience report, demonstrating the use of authentic learning examples from the immediate student environment, presenting a specific example of a mobile assessment tool. We demonstrate how this tool can be used to provide multiple, authentic learning examples for a range of Software Engineering topics, from introductory to advanced topics.

RESULTS

In this paper, we demonstrate the development of the PracMarker teaching tool - a mobile application used to manage laboratory-based assessment - and its use as an authentic example throughout a range of introductory and advanced courses. Students are exposed to the PracMarker teaching tool from their first week of studies, and quickly become familiar with its operation and limitations, making it an ideal example for later courses. We are able to leverage the design and architecture of the PracMarker teaching tool to support the development of authentic examples for a range of Software Engineering topics; in this paper, demonstrating examples and exploring the student experience for data modeling, mobile and wireless networks, computer and network security, distributed systems and software architecture. Our multiple examples illustrate not only the wide range of authentic learning examples that may be extracted from these kinds of tools, but also the process of learning that results from such integration.

CONCLUSIONS

Our initial work with this tool-as-tool has very quickly identified the gaps between a student’s first statements of “Well, surely, we can do...”, their transition to a more constructive “perhaps if I try…” and their mature comprehension of the issues raised by the question, in this context, leading to statements such as “within these constraints, and to solve this problem, I would undertake these actions”. We do not present PracMarker as a “gold standard” technology for first-year practical assignment marking but as a valid and authentic development procedure that we have been through and that, warts and all, we expose to the students to efficiently provide an authentic experience to contextualise and extend their knowledge.

KEYWORDS

Teaching tools, authentic learning experiences, software engineering

Introduction

Authentic learning experiences require students to solve a real-world scale problem in an authentic environment that elicits professional behaviours. One key characteristic of authentic learning is the inclusion of *authentic learning examples*, example problems and exercises that help students to see the relationships between the theories and concepts that they are taught and the problems that they must solve as a professional (Herrington & Kervin, 2007) (Herrmann & Popyack, 1995). Authentic learning builds upon the theory of *situated learning* (Singer & Willett, 1993), where 'learning occurs most effectively when [...] learners engage in authentic activity with community members'. Unfortunately, a significant problem in designing authentic learning activities is in designing activities that are indeed authentic but whose motivations and context can be easily understood by students, and in developing examples of sufficient complexity to illustrate real-world levels of difficulty.

Students are engaged and motivated to participate if they are involved in solving a real-world problem. However, students are not often capable of the levels of abstraction or maturity required to make the best use of the real-world problem environment. Assessment requirements dictate the careful selection of real-world problems that have a known completion time or, when multiple problems are used across different groups, defined levels of difficulty. Project-type courses readily integrate industry-associated projects and other authentic problems, but this may still not be of benefit to the majority of students.

Authentic activities used within the classroom are necessarily different from those that occur in industry - students are constrained in the time and resources available for their projects, and require scaffolding in their development. However, authenticity need not require a constant stream of site visits or industry-run projects. Authenticity can be gained in many ways, from the traditional view of real-world problems, to the, more practical within an academic sphere, examples, such as a student's solutions to problems from their own lives, students' own projects, as well as using real-world data from industry or scientific projects (Lee & Butler, 2010) (Lombardi, 2007). It has been shown that students respond to the use of authentic examples from their everyday environment in order to build enthusiasm and lifelong learning (Linn & Muilenburg, 1996).

How can we provide the benefits of an authentic learning experience to all student groups, when we must consider the impact of limited teaching resources and increasing class sizes? We could focus on industry case studies and provide a simulated industrial environment but we risk trading authenticity for ease of implementation. We are, however, ignoring the lessons that we can provide from the problems that we face as educators in an increasingly tool-based teaching environment. Contemporary teaching activities often require a large percentage of faculty to be involved in the development or selection of learning management systems, teaching tools and teaching support systems as part of their day-to-day work. The staff are immersed in an authentic design, selection and production environment that is, effectively, shielded from the students.

Within the Software Engineering domain, we need look no further for an authentic learning context than the learning environment of our own students, and the design of the numerous, complex information management and software systems that exist around them. Exposing our students to the processes that we employ to develop these learning activities and contexts provides a degree of authenticity without the need to identify and develop domain expertise in an external or industry-related context. Students are able to explore their software design and development skills while being able to access expert knowledge and can contrast their solution models with those implemented in the `real' systems, building questioning and analysis skills.

In this paper, we present the PracMarker teaching tool, a practical solution to an existing teaching problem, and also a teaching tool that we use to discuss sound and authentic practices with our students. We will discuss the way in which we use it to solve an existing problem and to add authentic elements to the teaching of introductory and advanced Software Engineering topics. As we will show, we can gain a great deal by providing a real design problem, with its solution, and allow students access to the final tool to carry out environmental testing.

Related Work

Authentic learning involves solving real-world, complex problems using techniques such as case studies, problem-based learning, role-playing and virtual communities. Authentic learning environments are not confined to the immediate problem at hand, or the concept that it is intended to be learnt, instead they extend - like real-world problems - to encompass other, related problems and disciplines. The use of such learning examples, ranging from individual authentic learning activities through to full immersion in professional practice, provides increased motivation, engagement and contextualised learning (Lombardi, 2007). The basis of situated learning techniques, such as authentic learning, is that learning cannot be abstracted from its context: it is imperative to include authentic examples and contextualised examples to provide opportunities for deeper learning. There has been a great deal of work on providing authentic learning activities and environments for our students. Such work has typically taken the form of either industry case studies, or the use of industry-standard software tools to illustrate professional practice and industry concerns.

Shabo *et al* (1996) demonstrate early work in the use of research and industry examples to motivate students in the area of Computer Graphics. They illustrate a situated learning technique of apprenticeship whereby students are exposed to problems, real-world examples and case studies, while able to reflect and discuss their progress with discipline experts. Zilora (2004) introduces an industry standard software engineering project based around the use of web services to support the business operations of a fictional company. The use of industry standard development tools, i.e. XML and Java WebServices, enables students to gain relevant experience, although the context for their study is still far removed.

Phoha (2005) introduces the use of real-time embedded operating systems, which are smaller and easier to model and understand than full-scale operating systems, to teach task scheduling, process management and fundamental operating system design. Gasper *et al* (2008) discuss the use of industry standard Linux operating system modules as the basis for a series of projects in operating systems and computer networks courses. Tedford (2003) talks about the use of current events as a source for learning activities for a data structures course. Buchele (2009) discusses the use of the One Laptop Per Child programme as an example of a modern computer systems design challenge, which can be used as a source of learning examples for computer systems, computer architecture and organisation and computer networks courses. In addition, this program can also be used as a source for the discussion of ethics and professional behaviour. Fife (2010) discusses the use of undergraduate research projects as a way of embedding authenticity into the curriculum. The use of real research projects, often incorporating multiple disciplines provides a suitable environment for authenticity.

Teaching Tools

In this paper, we demonstrate the development of the PracMarker teaching tool - a mobile application used to manage laboratory-based assessment - and its use as an authentic example of Software Engineering throughout a range of introductory and advanced courses. Students are exposed to the PracMarker teaching tool from their first week of studies, and quickly become familiar with its operation and limitations, making it an ideal example for later courses. The structure of the PracMarker system may be seen at Figure 1. The PracMarker tool development was driven by a real requirement. Studying the design and development of this real tool provides a case study that is authentic because it is situated in a real environment, reflects genuine user-requirements and went through a real development cycle.

This can then be presented to students as a far more useful case study than those usually possible in small-scale and synthetic environments.

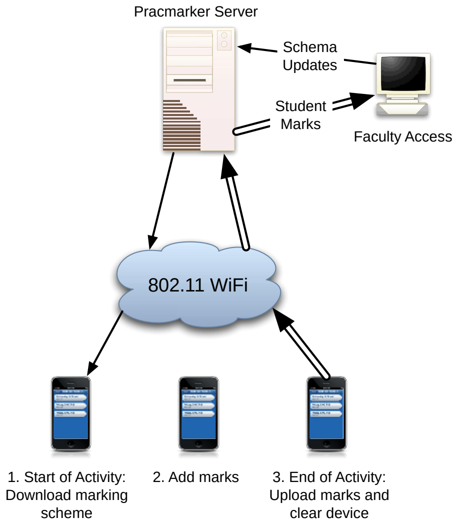

Figure 1 The PracMarker System

The PracMarker System

We first identify why we designed the PracMarker system: to enhance our capabilities in the timely and accurate capture and confirmation of student assignment marks in first-year programming assignments. These assignments are often quite straight-forward in nature as we seek to develop key skills in specification, design development, planning and problem solving, along with developing the skill of programming. To maximise the benefits of these exercises, in-laboratory demonstrators are frequently used to provide immediate feedback and assessment when students present their work. This ensures that students do not accidentally provide a correct answer using incorrect technique and is far more flexible than an automated system. However, having multiple laboratory sessions, many students and multiple demonstrators active at the same time increases the complexity of a relatively simple exercise, marking students and collating these marks for inclusion in assessment.

Prior to the development of commodity hand-held devices, many marks were recorded on simple paper forms, pre-printed to show marking ranges, student numbers and to provide useful information to the marking team. One set of these paper forms was shared between the markers within a single laboratory session, introducing problems of concurrent access but reducing the potential for inconsistencies that could easily result from multiple data sources. While electronic systems can be very useful in this space, the mobile nature of a marker and the finite number of seats in a laboratory conspire to prevent us easily allocating a ‘marker's terminal’. With this model, we still have the issues of concurrent access, and we introduce an intermediate mark recording step, to encompass the period of time when taking the mark from the student to the marking terminal.

Paper forms have many advantages for large group marking. These include the natural resilience of a record on paper, the guarantee of consistency if a single marking form is used for one practical and the tangible feedback of seeing the mark recorded. However, a single form provides a bottleneck for multiple markers, needs to be physically transcribed to assessment records and can, ultimately, be lost - rendering its many desirable physical properties moot. While many management issues can be dealt with via staff training, the bottleneck/consistency trade-off will ultimately slow down marking and cause delays to students where the actions of other students could potentially stop a student from being marked. The obvious solution to the bottleneck, multiple paper forms, solves one problem but at an immediate risk to data consistency.

With the introduction of COTS hand-held mobile devices, it is now possible to have a marking system that meets our consistency requirements but increases our overall efficiency of marking and reliability of mark transfer. The PracMarker system was developed over a six-month period as a two-part project: a client constructed by an external three-person development team, and a server constructed in a collaborative development environment with local and external development teams. The client is an iOS App, built for iPod Touch devices specifically, and restricted to a defined set of devices. Figure 2 shows the PracMarker device interface, illustrating its utility in managing multiple data sets, and flexibility in recording student assessment information.

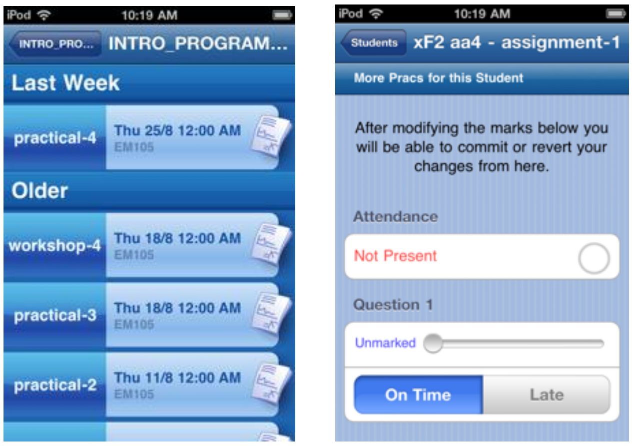

Figure 2 The PracMarker Interface.

The development of PracMarker raised a number of questions about marking methodology and student feedback. From the perspective of this paper, however, it provided a rich opportunity to capture our specification, design and development knowledge, in order to use the construction of this teaching support tool as a teaching tool in its own right.

The characteristics of PracMarker are:

- Client/Server Architecture
    - Central Definition: PracMarker assignment specifications are defined at the server as schema, transferred to the server by faculty, and downloaded to devices on demand. PracMarker devices only store assessment data temporarily until synchronisation, when assessment data is uploaded to the server. Schema are stored and defined on the server.
    - Synchronisation Barriers: PracMarker devices are designed in a client/server framework and must synchronise with the server to ensure that assignment specification information is consistent. If devices are not synchronised, then it is possible for two devices to record conflicting information on a student. This barrier is addressed by staff training, for reasons outlined below.
- Information Consistency
    - Rapid Update: PracMarker devices may be synchronised at any point to transfer marks to and from the server.
    - Tangible Update: Students may inspect their marks on the server immediately following an update. This reassures them that their marks have been recorded and immediately addresses the issues of `lost paperwork' raised previously.
- Ease of Use Considerations
    - Any Student, Any Prac, Any Marker: PracMarker allows students to be marked off for previous activities, although marked as late, or students from another session who are visiting may be allowed to be marked off in a different

session. A student's final mark for an assignment may come from a combination of marked components spread across multiple devices.

    - Easy Search: Students may be located by name, ID or by assigned group. Students may attend their own assigned activity or, for a variety of reasons, may attend another. Locating students quickly and easily reduces the amount of time wasted in assigning a student's mark and maximises the utility of practical demonstrator resources.

- Security Considerations
    - Secure Access: Only a defined group of users may use the PracMarker devices and access is modular by course. Authorised users may have their authorisation immediately revoked at any time from the server.
    - Sandbox: The marks uploaded to the PracMarker server are isolated from the grade entry systems that are used to record a student's final mark for course. While students can see the uploaded marks, these cannot be used for final grading unless a faculty member inspects and explicitly authorises the transfer. This oversight step sandboxes the uploaded marks and does not require us to provide privileged access to the much more sensitive and secure systems that store all of a student's grade components.

As such, PracMarker is a good example of a real-world data entry and management system, using a client/server framework. As we will see, the factors considered, questions raised and set of implemented functionality, provide a rich source of authentic software engineering discussions. PracMarker can also be set-up for ‘dummy’ assessments, based on a provided test environment called ‘TRAINING_WHEELS’. These training assessments are ignored for marking purposes, so students can work with the tool to test their understanding of the specifications and limitations against the tool's final implementation. By recording our problem, our specifications, our design processes, our interactions with the developers and our ongoing attempts to improve this tool, we are now capable of using the teaching support tool as a teaching tool itself.

Example Assessment Activities

We have a rich source of material from which to draw informative and useful assessment activities. These go beyond the fundamental design and lessons learned in constructing PracMarker, extending to much more complicated issues of data consistency and environmental awareness. The design constraints of a tool that need only be available 9-5, Monday to Friday, in a heavily supported University technology environment are very different from those of an embedded system deployed to Mars, yet the way in which we deal with our environment illustrates, very clearly, one of the major sources of authentic learning - a realistic reaction to a realistic environment.

In the following subsections, we touch on the learning aspects and assessment that can and has been derived from the PracMarker project. Each section begins with key questions, directly derived from the project, that are of relevance to the area. The questions give examples of how we would use this tool-as-tool in a lecture, tutorial or assignment-based interaction with students. A student can consider the question, look at the final system (with which they are very familiar) and, in many circumstances, then test their understanding by using the device in training mode if available.

Data Modelling: “How do I store assignment results for students?”, “What operations do I need to support for this set of data?”

The schema structure for data is extremely simple. Once a marking scheme has been defined and uploaded to a device, marks can then be entered for any student. Once synchronisation is completed, the marks are then immediately available to students without any manual post-processing delay. The marks are stored in Comma-Separate Value (CSV) format; we present this to the students as a sensible decision, because the ultimate fate of the marks is to be used in assessment decisions. We never need to sort on these component marks, nor are they used for mark-up or as part of larger table formations, hence RDBMS technology is unnecessary. Marks are stored as components within an assignment, found within a course, found within a given semester and year. This provides unique identification of a mark and association with a student, as defined by faculty staff. Once synchronisation has occurred, the data stored on an individual PracMarker device is removed, leaving only current assignment specifications.

Multiple markers may assess a student during a single or multiple sessions. In our first year courses, we encourage students to request assessment as often as they require, with the understanding that the feedback from an earlier assessment will be valuable in aiding their understanding, and their ability to complete the required assignment work to a high level. Accordingly, the assessment records for an individual student may be divided amongst multiple entries and across multiple devices. Within an assignment, marks are stored as a sequence of uniquely identified CSV files, coded with the time of upload, the device ID and the marker ID. On the server, these files are processed according to a configuration specified policy - for example, a student may be awarded the maximum mark for each individual question, or they may be awarded the last recorded mark for each individual question.

Mobile and Wireless Networks: “Which is better: fixed routing or mobile routing?”, “What are the real world constraints for wireless networking?”

The devices used can employ WiFi 802.11 and Bluetooth. The design of PracMarker was for the standard usage environment, which was a small number of devices in an area serviced by WiFi. Thus, PracMarker devices are designed to work on WiFi as a relatively long-range, reliable technology for delivering the marks back to the server. This is a fixed-routing strategy, where the WiFi client is associated with a (set of) WiFi base stations and very little dynamic routing takes place. Given that practical marking takes place in a given room at a given time, for the vast majority of cases, producing a complex dynamic-routing protocol that flips interfaces on demand is a waste of time and effort. The answer to the question, *“Which is better: fixed routing or mobile routing?”*, is, as always, it depends, but the existence of a familiar device that actually does this makes it far easier to ground student knowledge.

At this stage, the Barrier Synchronisation limitation allows devices being used in the same assessment activity to get out of synch unless synchronisation is carried out regularly. Why? By asking this question, we force students to consider the difficulty of setting up multiple associations of Wireless networking access points, especially where authenticated Access Points are used to move data to the server. Students are also asked to consider the efficiency of a Peer-to-Peer (P2P) implementation of the same system, where one or more of the devices maintains a WiFi uplink and uses Bluetooth, for example, to communicate with the rest. By considering these aspects, which are fundamental design questions, in conjunction with the requirements for the standard operation of the system, students assess the benefit of each approach and the return gained for the greatly increased complexity of an active P2P routing solution.

Computer and Network Security: “What happens if an iPod gets stolen?”, “What about Denial of Service attacks?”

The isolation of the server from any formal marks systems provides an opportunity to discuss levels of security, degrees of trust and the use of sandboxing and clearing-house models. The model that allows us to revoke access for an individual device, if lost or stolen, or a set of users, where our ability to perform these actions only hinges on access to the server, clearly illustrates the control that we can derive from a client/server model. The system, as implemented, is prone to a small set of Denial of Service attacks but the environment in which it is used lowers the risk substantially, as an intruder would need the correct subnet access, the correct authentication codes (from a stolen account) and would also need to be able to generate enough data to fill the marks storage disk, which would require a long period of access or a large number of devices. This allows us to justify our control model that restricts access based on time and week specifications built into the schema - an intruder would have to be an insider, or exceedingly well-informed, in order to successfully attack the system. The sandbox structure allows a faculty member to clean the data prior to upload. This simple design illustrates layers of protection, degrees of trusted access and most-likely scenario analysis in a way that does not compromise access to restricted systems!

Software Architecture: “Is this Client/Server architecture or State-Logic-Display pattern?”, “What communication patterns are needed?”, “How can we change data models or communication requirements?”

One of the key issues in any course on Software Architecture is to provide students with an example software architecture that is complex enough to exhibit real-world architectural issues, while also encompassing software requirements that are understandable to a typical student. The PracMarker system adopts a Client/Server architectural style, with synchronous communication requirements between an unlimited number of Clients and a single Server. Real systems exhibit architectural variations and refinements that can make identification of the relevant architectural style difficult. Students are able to construct an architectural model of both the components (devices and server) within the system, and the connectors, modelling synchronous uploads between the server and the devices. They are able to compare their models with the PracMarker system to obtain clarity and perspective.

We are able to ask students questions regarding the suitability of the architecture in response to changes in requirements, such as changes to the data model - *can we change the way that the data is stored independent of the interface design?* It is easy to say to students that a key requirement of a Client/Server architecture is that end-user (device) processing should be limited to data entry and presentation. However, it is only when they try to modify this, for example, to maintain a consistent view of student data on all devices in addition to the server, that they realise why this is of benefit to the design of the system.

Concurrency Management: “What if two pracmarkers try to change the same student's mark at the same time?”, “Does information ever go stale?”, “How long do we wait for access to a shared resource?”

Data consistency is a complex problem, given that it is possible for a student's marks to be spread across multiple devices. Staff training solves the majority of these issues by taking advantage of the rapid synchronisation process to keep the devices updated. How do we choose to resolve conflict? If we allow a “last update sticks” approach, then a student's mark is potentially only one late update away from changing. If we ignore zero entries, how do we reduce marks later if an error was made in the data? In the existing system, we have adopted a simple approach to conflict resolution where the maximum mark is recorded where conflict occurs. We have taken this approach as it is possible for students to have their worked marked multiple times and by different people.

Conclusions

Our initial work with this tool-as-tool has very quickly identified the gaps between a student's first statements of “Well, surely, we can do...”, their transition to a more constructive “perhaps if I try” and their mature comprehension of the issues raised by the question, in this context, leading to statements such as “within these constraints, and to solve this problem, I would undertake these actions”. Students can understand the valuable lesson that design issues are often dependant upon one another or otherwise related, if we frame these issues inside a truly authentic framework.

We do not present PracMarker as a “gold standard” technology for practical assignment marking, although it performs very well for our school, but as a valid and authentic development procedure that we have been through and that, warts and all, we expose to the students to provide an authentic experience to contextualise and extend their knowledge.

Real tools are being developed to support teaching on a regular basis and we recommend taking the opportunity to expose students to the authentic design and decision making process that is used to construct these. Addressing the implementation issues that arise helps to give every student a deeper understanding of the many pitfalls implicit in software design and tool construction.

References

Buchele, S. (2009). USING OLPC LAPTOP TECHNOLOGY AS A COMPUTER SCIENCE CASE STUDY TEACHING TOOL. *Journal of Computing Sciences in Colleges, 24*(4), 130-136.

Fife, L. (2010). Authentic Learning through student research projects. *Journal of Computing Sciences in Colleges, 25*(4), 37-43.

Gaspar, A., Langevin, S., Armitage, W., & Rideout, M. (2008). ENABLING NEW PEDAGOGIES IN OPERATING SYSTEMS AND NETWORKING COURSES WITH STATE OF THE ART OPEN SOURCE KERNEL AND VIRTUALIZATION TECHNOLOGIES. *Journal of Computing Sciences in Colleges, 23*(5), 189-198.

Herrington, J., & Kervin, L. (2007). Authentic learning supported by technology: ten suggestions and cases for integration in classrooms. *Educational Media International, 44*(3), 219-236.

Herrmann, N., & Popyack, J. (1995). Creating and Authentic Learning Experience in Introductory Programming Courses *Proceedings of SIGCSE'95* (pp. 199-203).

Lee, H.-S., & Butler, N. (2010). Making authentic science accessible to students. *International Journal of Science Education, 25*(8), 923-948.

Linn, M. C., & Muilenburg, L. (1996). Creating lifelong science learners: what models form a firm foundation? *Educational Researcher, 25*(5), 18-24.

Lombardi, M. (2007). *Authentic learning for the 21st century: an overview*: Educause Learning Initiative, ELI Paper 1:2007.

Phoha, V. (2005). A SIMPLE REAL TIME EXECUTIVE tinyexec AS A TEACHING TOOL IN AN OPERATING SYSTEM CLASS. *Journal of Computing Sciences in Colleges, 20*(6), 92-98.

Shabo, A., Guzdial, M., & Stasko, J. (1996). Addressing Student Problems in Learning Computer Graphics. *ACM SIGGRAPH Computer Graphics, 30*(3), 38-40.

Singer, J. D., & Willett, J. B. (1993). Lessons we can learn from recent research on teaching: it's not just the form, it's the authenticity *Proceedings of the Symposium on Innovations in Teaching: How Can We Improve the Communications Skills of Students? Annual Meeting of the American Stastical Association*.

Tedford, P. (2003). USING CURRENT EVENTS AS A TEACHING TOOL FOR AN UNDERGRADUATE DATA STRUCTURES COURSE. *Journal of Computing Sciences in Colleges, 18*(4), 50-55.

Zilora, S. (2004). Industry-Based Web Services Project as a Classroom Teaching Tool *Proceedings of the 5th Conference on Information Technology Education* (pp. 13-18).

Copyright statement