

Interactivity and Integration in Virtual Courses

Claus Pahl

Dublin City University, Dublin 9, Ireland

cpahl@compapp.dcu.ie

Abstract

Web-based virtual courses focussing on content delivery only have turned out not to be as successful as expected. We will investigate reasons and suggest some remedies – using our own virtual Database course for illustration. Interactivity and integration are the two main remedies, which shall lead towards improved, learning-oriented virtual course environments. We will introduce a formalism to support their development and analysis.

1. Interactivity and Integration

In virtual courses traditional educational activities or services such as lectures, tutorials and lab sessions are replaced by Web-based counterparts. In this paper we will discuss the design and evaluation of interactive integrated multi-service virtual courses. Despite potential advantages of Web-based delivery [1], there are also risks involved. In order to support active learning virtual course systems need to provide more than a static representation of knowledge. Interactivity is a means for engaging and encouraging a student. Observing student behaviour, we can see that students typically use different activities with different regularity, e.g., a ‘just-in-time’-learning in labs close to tests/exams vs. a more regular lecture attendance. Virtual courses need to be made more *interactive* or engaging. The realisation of educational services such as lectures or tutorials should be fully *integrated* in order to exploit the potentials of the new technology such as availability and integrated access. Both Web-based systems and study habits might lead us away from a successful learning experience, i.e., the usability of the system, success in exams, and user satisfaction. In order to overcome these shortcomings, we propose *integrated interactive courses*. Integrating interactivity allows students to learn actively – a necessity if practical skills have to be taught in a course – integrated with more conceptual parts of a course. Our hypothesis is that students learn by doing – which is true for most science or engineering subjects – and by reflection. To teach computer programming skills, as in our

own virtual course [2], requires in particular interactivity support. *Interactive* elements are typically accompanied by theory. Interactivity might be supported by guided tours where topics are demonstrated and by elements where the student can freely train skills. These forms - corresponding to lectures, tutorials and labs - should be integrated, e.g. supported by an adequate navigation structure allowing the student to access complementary material, integrating learning activities around course topics.

We shall focus on techniques to support the design and evaluation in a coherent framework. In order to formulate the structure of a multi-service course and the behaviour of students within this structure, we suggest a reference architecture for the integration of educational services. A notation based on this model will be introduced that allows us to describe behaviour based on this architecture.

2. A Model for Integrated Interactive Services

When a multi-service virtual course has to be developed an in-depth understanding of the course structure and the expected dynamics within the system is essential. We will present a model of a layered reference infrastructure and a notation to express the dynamics in such a course. It allows the developer to formulate expected behaviour in the design phase, but also to analyse student behaviour during and after delivery. A suitable formal approach is sought to capture the static dependency issues and the navigation infrastructure, which requires means to express the sequential but also concurrent use of services. A second criterion is a notion of state in order to remember the usage history (which pages have been visited). We use Petri-nets in combination with a notation called path-expressions that can express user behaviour over time.

In a system with fully integrated educational services, the student could interrupt using the lecture service, practice relevant skills using the lab service, and then resume the lecture. In a virtual system, changing between modes of learning can easily be supported by appropriate navigation support connecting the services. The services can be represented in layers. The first layer is the lecture service, the second layer represents interactive services:

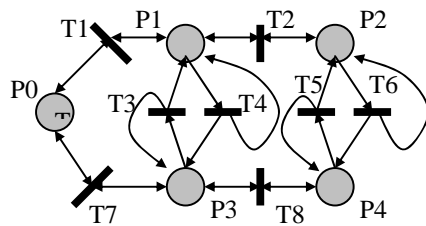


Figure 1. Virtual course topology

Usage patterns defined by the developer are based on causal dependencies and can describe the intended use. Path expressions formulate the patterns. The expression $P1;(P1||P3);P2$ means that P1 shall be accessed first, then P1 and P3 are looked at in parallel (link T4, Figure 1), and finally page P2 is visited. Besides the sequence operator ‘;’ and the parallel composition ‘||’, other operators such as choice $P1|P2$ (choose between P1 and P2) and iteration $P1^*$ (visit P1 any number of times) are available. These path expressions, formulated on the defined course topology, describe the expected usage and should be implemented by a respective navigation infrastructure. The expression $P1;(P1||P3);P2$ can be interpreted as a guideline for using the system: look at lecture page P1 first, then open tutorial/lab page P3 in parallel and practice, then go to the next lecture page P2. The formal model also allows the teacher to capture actual usage of the system for the course evaluation in terms of the notation provided, see [3].

Our own virtual database course [2] replaces traditional lectures with a lecturer’s speech and overheads by audio and synchronised visuals. The audio presentation controls the presentation of visuals. Our course system improves the classical delivery by making it more personalised and more interactive through individualised delivery. The student can interact freely with the lecture service by controlling the audio stream. A tutorial service illustrates the lecture material dynamically. It guides the student through a series of examples with increasing complexity. The tutorial service allows the student to practice material learned in the lectures. The system offers also individual feedback in form of a check-function. A laboratory service provides practical material beyond the examples used in the lectures. The integration of these services is essential. An infrastructure allowing students to navigate between corresponding parts of different services needs to be provided. The topology model allows us to formulate this infrastructure. Using the path expressions, a developer can specify behaviour which is expected and which shall be supported in the system by the navigation infrastructure.

3. Evaluation and Conclusions

We will address evaluation here only in order to validate and justify the ideas and concepts that have been presented

– more can be found in [3]. We have been looking at *student behaviour* in two dimensions: usage in time (frequency/regularity of usage, number of accesses) and usage in space (usage patterns based on the topology). These evaluations are based on the topology model, which forms the backbone of our approach. A look at usage patterns shows that the lecture service is typically used without accessing the interactive parts, but when lab or tutorial services are used, students do look up definitions, concepts and examples in lectures. Here, a close integration of static and interactive services is useful. The analysis of frequency and regularity of system usage shows that lectures are used on a more regular basis than tutorials and labs. The majority of students leave exercising until close to tests and exams. An improved integration of interactive elements into the lectures might have led to a more regular use of tutorials and labs, but the integration has certainly improved the usability of the interactive services themselves. We have monitored the *student success* in exams over a period of three years. There is no difference between traditional and virtual delivery with respect to performance in questions about topics supported by interactive services – our last delivery with improved interactivity showed even better results. The same is true for an overall comparison between traditional and virtual delivery [4]. Two usability criteria emerging from *student surveys* are the need for interactive elements that provide good feedback to student input and the need for a good integration and navigation structure. Active learning and feedback are important elements of a course, see e.g. [5].

Our main objective has been to support the development and delivery of virtual courses, providing educational activities equivalent to those in traditional courses. Interactivity is a desirable feature in virtual courses; the integration of interactive and other parts is essential. Our formal model and the notation allow us to describe infrastructure and behaviour in integrated interactive virtual courses. Model and notation provide a tool for teachers and developers to design and analyse advanced virtual Web-based courses.

References

- [1] J.A. Lennon. *Hypermedia Systems and Applications*. Springer-Verlag, 1997.
- [2] C. Pahl. CA218 Databases (Course Web Site), 2000/2001, URL for Web page: <http://www.compapp.dcu.ie/~cpahl/>.
- [3] C. Pahl. The Evaluation of Educational Service Integration in Integrated Virtual Courses. *IEEE SAINT Workshop on Internet Supported Education*. IEEE Press, 2001.
- [4] A.S. Smeaton and G. Keogh. An Analysis of the Use of Virtual Delivery of Undergraduate Lectures. *Computers & Education*, 32(1):83-94, 1999.
- [5] American Association for Higher Education. *Seven Principles for Good Practice in Undergraduate Education*. <http://www.aahe.org>. 1997.