World Library and Information Congress:
69th IFLA General Conference and Council

1-9 August 2003, Berlin

Code Number: 037-E
Meeting: 87. Audiovisual and Multimedia
Simultaneous Interpretation: Yes

Teaching the use of publicly-available formats for multimedia as part of a library school curriculum: the vision and the reality

James M Turner
professeur
École de bibliothéconomie et des sciences de l'information
Université de Montréal
CP 6128, succursale Centre-ville
Montréal, QC CAN H3C 3J7
james.turner@umontreal.ca
http://mapageweb.umontreal.ca/turner/
telephone: +1 514 343 2454
fax: +1 514 343 5753

Abstract

After several years of using proprietary software in teaching a multimedia course in a library school, several factors that came together in the fall of 2002 offered a chance to make the leap to teaching open, publicly available recommendations of the World Wide Web Consortium as tools for building multimedia information systems. The use of public rather than proprietary tools seems to be the most desirable route to take since it assures longevity to multimedia information and promotes standardised methods for building systems for housing the information. However, adopting such an approach involves a number of difficulties arising from the public nature of the tools. In this paper we expose the reasons for adopting this approach, describe developments in the publicly-available tools for building Web-based multimedia systems, discuss some of the difficulties encountered in using these tools in the context of a course in a library school, and evaluate whether the advantages of using public software tools in teaching outweigh the difficulties associated with them.

Introduction

By the mid-1990s, computer technology had advanced to the stage at which the need for a course in multimedia information systems was felt in our school (École de bibliothéconomie et des sciences de l'information). However, offering such a course was problematic because the course began at a time when severe budget cuts came into effect at our university (Université de Montréal). Although the school has a well-equipped computer laboratory, it was impossible to justify the purchase of specialised equipment and software for teaching a single course, especially one that students were not required to take. At the same time, we felt that it was of critical importance that the students get some hands-on experience in order to understand the theory, so that access to technology and equipment was necessary. Accordingly, the first time the course was offered, we obtained permission to use a well-equipped multimedia laboratory in the Education Department at the other end of the campus. Two students enrolled and the course was given as a tutorial. Using HyperCard (Apple Computer), the students built a prototype for a multimedia information system about the last few elections held in Québec and Canada. As basic criteria, the system had to include text, still images, sound, and video. These criteria are still a requirement for the course.

Now the course is giving using the school's own computer lab. The main activity for the course is building a prototype for a multimedia information system. This approach permits discussion of the theoretical aspects (conception, structuring the system, design issues, interface, navigation and orientation, user understanding of the system) and provides the opportunity for hands-on experience which promotes understanding of the theoretical issues. Students work in small groups (minimum two, maximum three), and they are encouraged to choose the topic for their prototype. The topic is approved as long as it has to do with building an information system.

Twice this formula has changed because of special circumstances. In 1996, the twelve students enrolled in the course worked on a common project. The director of the school was mandated to investigate the possibility of building SLISnet, an information system for library schools worldwide. The course was organised to give the students the opportunity to work as a single multimedia development team, for a real client. In 2002, the eight students enrolled in the course worked as a single team to build a Web site for the MetaMap, again with a real client. We will discuss the MetaMap in more detail in the sections following.

Information technology changes very rapidly. Computer hardware and software, and specifically software for constructing multimedia products, are of course part of this. Because of this situation, it has been quite a challenge to offer the course (BLT6336 Multimédia et hypermedia, in its most recent incarnation). Our inability to justify the high costs has meant that we have never been able to buy the expensive high-end software used in the industry to build multimedia products. In any event, this too is constantly changing, and from the beginning we felt that the really important issue is to teach the theory, which the students will then be able to apply to any software that comes along. But we also insist on hands-on experience with building a prototype system in order to get a grasp on the theory. Thus, each year the course has been offered, a decision has been made during the summer, according to the latest information available, about which software was available free or cheaply enough for

us to install it in our computer laboratory in time for the course to be offered in the fall term. This has always been a source of stress, because it is difficult to plan. Once the course has started, it is necessary to live with our decisions, whatever the consequences. Some faith and trust on the part of the students is also required, and this is sometimes destabilising for them, in spite of the many reassurances we try to provide.

Over the years the course has been offered, the World Wide Web has also been developing at a rapid pace. We have observed the shift of multimedia information from local computer systems, beyond CD-ROM publishing, to Web-based multimedia information systems. It is obvious now that these systems will exist more and more in the networked environment and not as isolated products. Thus, we have committed to using the Web as the platform our students use for developing their multimedia information system prototypes.

There are other important reasons for adopting this approach. Our school places strong emphasis on teaching structured documents, markup languages, using valid code, and the interoperability of files across platforms. In the first year of our master's programme, the students are required to learn html and to build a Web site with valid code. Using the Web as the environment for building a multimedia system in the context of an elective course offered in the second year of the programme allows the students to build on their skills and to acquire new ones. Since the Web is already a familiar environment for the students, it is easier to teach the theoretical notions of multimedia information systems in ways they can understand. In addition, examples of both good and bad systems abound on the Web, so that many examples of points we wish to cover are available for them to see.

Perhaps the most important reason of all for adopting a Web-based approach is that tools are becoming available which will allow us to ensure the longevity of the information we wish to disseminate in a Web-based environment. Every time a new version of a piece of software or an operating system is published, we are forced to buy these tools all over again, even though the ones we already have work adequately, because soon it becomes impossible to interact with other users if we do not have the latest versions. All of us live with this problem. But a worse problem is that over time, documents created with older software are not able to be read by newer software, so information becomes lost. Standards sponsored by ISO or by the World Wide Web Consortium and a number of other players are very helpful in this regard because they are independent of the software and hardware. Examples of these are XML, HTML, and JPEG. We try to instill in our students the advantages of choosing publicly available tools over proprietary software.

However, proprietary software usually seems more attractive because it is often faster to learn, has a friendlier user interface, and offers more features than publicly-available tools do. This is due to the nature of the environments in which each type of tool is developed. In the private sector, a lot of development funding can be put into a product that promises to pay off in the marketplace, and since competition is keen, any new technological advances or software features developed are marketed as quickly as possible. Public tools are necessarily developed more slowly because they involve their user community in a much deeper way, seeking inputs from users in response to a request for comment, holding meetings among the stakeholders to

discuss the issues, setting up collaborative testing among users, refining the norm according to all these inputs, and so on. These processes are necessarily time-consuming. However, they pay off handsomely in the long run because the end product corresponds to user needs, the product has been thoroughly tested, and longevity of data created using the norm is much more likely to be assured.

Making the leap

In the fall of 2002, several factors came together which permitted us to make the leap to teaching open, publicly available recommendations of the World Wide Web Consortium as tools for building multimedia information systems. There were new published versions of the Synchronised Multimedia Integration Language (SMIL), of the Scalable Vector Graphics (SVG) recommendation, and the Portable Network Graphics (PNG) format which offered improvements on previous versions and which thus had matured enough to be usable in a difficult classroom setting. In addition, some editing tools for SMIL and SVG were becoming available. Perhaps most importantly, for the first time a research assistant was available to do the background work, study and list the resources available, prepare tutorials for learning to code SMIL and SVG, help work out the protocols for hands-on practice, and to assist in guiding the students through this new territory.

The number of students enrolled in the course was small enough (eight) to permit working on a common project. The desire to have them do so was based on the fact that a learning tool we had been developing over several months for studying metadata sets, entitled the MetaMap (MétroMéta in French, MetaMapa in Spanish, and MetroMeta in Portuguese) was almost ready for publication. We wanted to have an interactive Web site for the MetaMap (http://mapageweb.umontreal.ca/turner/) so users could learn about it and how to use it. So we had a real client with a real need, and we sought to involve the students in the creation of the information system we needed to explain the MetaMap to users.

The MetaMap itself was created using SVG and involved an investment in understanding this W3C recommendation, learning how to read and write the code, finding an editor that permitted drawing the MetaMap and programming the various interactions we wanted it to include. The leap into such territory involves dealing with products which are only in the development stage, living with a number of unresolved ambiguities, and accepting that some aspects do not quite work as they should.

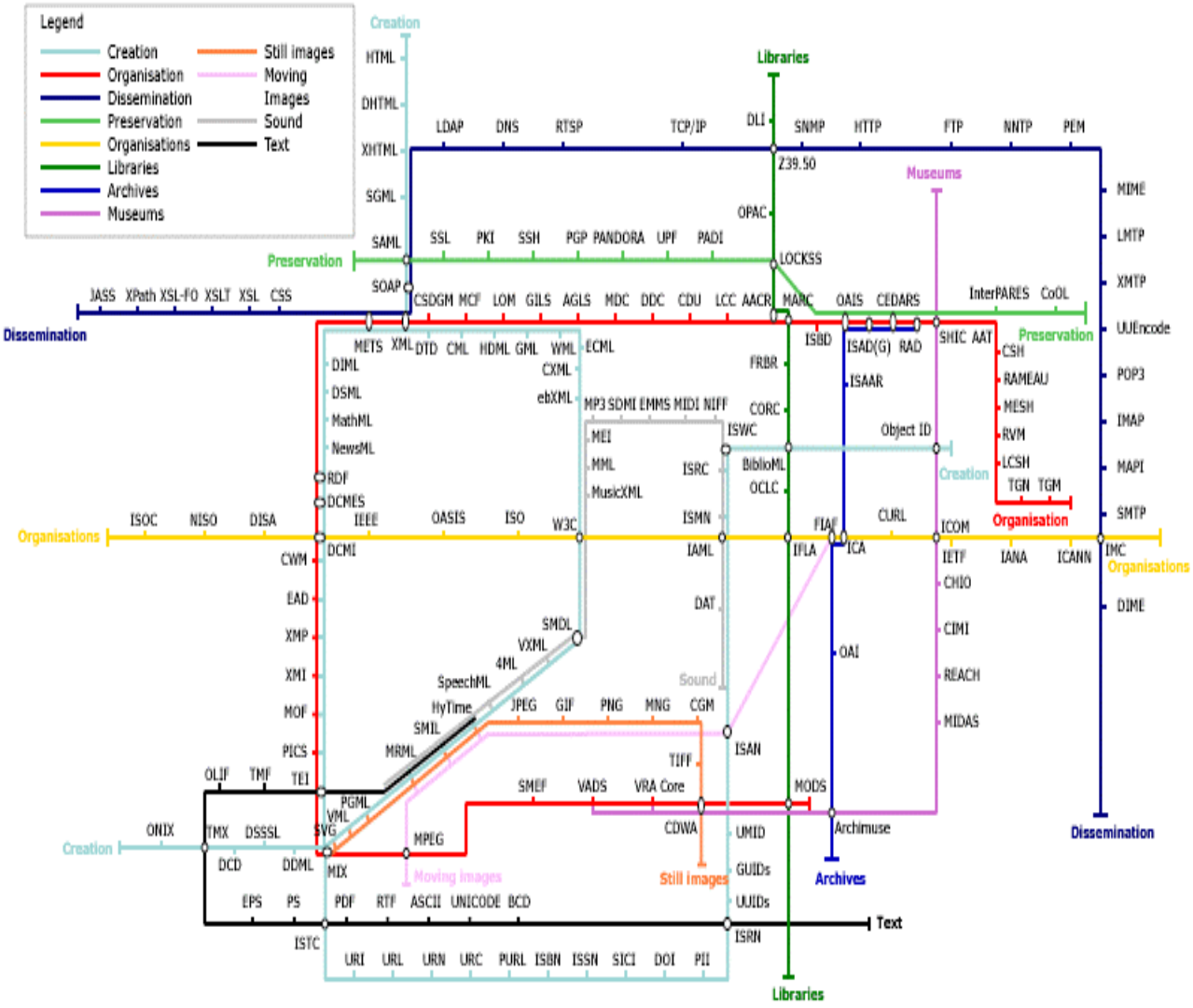
About the MetaMap

There are now so many metadata initiatives that it is difficult for Web users to gain a clear understanding of what they all are and what their role is in the broader library and information science picture.The MetaMap was created as a learning tool, and it is an attempt to sort out the very many efforts worldwide in working out norms and other information about metadata sets. The goal is to help people understand how the various metadata initiatives are related to one another and to information studies. The present version of the MetaMap includes almost two hundred metadata standards, initiatives, or sets. They are laid out as stations on lines of a subway map,

and the metaphor is that the map helps users navigate through this "metaspace", the universe of metadata.

[Figure 1: The MetaMap]

The MetaMap is interactive. The online version in English is at
http://mapageweb.umontreal.ca/turner/meta/english/metamap.html



The subway lines represent themes. One set of lines covers the processes of information management: Creation, Organisation, Dissemination, and Preservation. Another set covers institutions with expertise in information management: Libraries, Archives, and Museums. Another set of lines covers the various types of digital documentation: Text, Still Images, Moving Images, and Sound. There is also a special line, which occupies a central space in the map, for including organisations deeply involved in Web activity and metadata norms, such as the World Wide Web Consortium, OCLC, the IETF, the IEEE, and so on. Themes that intersect are represented as nodes in the network. For example, where the lines representing Libraries, Archives and Museums cross the Organisations line, the nodes are respectively IFLA (International Federation of Library Associations and Institutions), ICA (International Council on Archives), and ICOM (International Council of Museums). Similarly, elements that have repercussions in a number of areas are common to several lines. For example, SMIL, the Synchronized Multimedia Integration Language, is common to the lines representing Text, Still Images, Moving

Images, and Sound.

But the world is not an orderly place, and the elements in the MetaMap do not all fit in to the structure as neatly as we would wish. However, much effort has been put into showing the most important relationships. In addition to the nodes created where concepts intersect, within each line the various metadata initiatives are grouped together logically where this is possible. Thus classification schemes appear side by side on the Organisation line, XML and its derivatives appear together on the Creation line, and so on.

The Web versions of the MetaMap exist in English and in French. With the help of volunteer collaborators, we hope to build versions in other languages, so that the use of this educational resource can become more widespread. We are working out protocols for doing this, as well as for the upkeep and managementof the resource. At the moment, we have secured partners for preparing the Spanish and Portuguese versions of the map.

The vision and the reality

In this section we expose some of the difficulties involved in teaching publicly-available standards for building Web-based multimedia information systems. The vision of publicly-available tools is that they offer uniformity, standardisation, stability, and longevity. The reality is rather different, because of the development environment, although in the long term the highly desirable characteristics that make these tools attractive will be realised. The development environment of public norms involves initiatives by individuals with sponsorship of credible institutions, request for comment once the tool has been developed to the point where it can be considered at least minimally stable, feedback from users in a position to undertake such an endeavour, long meetings and endless discussions about the mechanics and the application to a wide variety of information objects, collaboration with individual and institutional testbeds to actually try out the tool, harmonisation of the needs and requirements of variety of users, and eventually forging a norm that is good enough to achieve recognition from an international standards body.

Private development of tools, on the contrary, is much less collaborative and much quicker. Small teams of bright developers working in secrecy can get something up and running and off to market in a relatively short period of time. However, the need to keep ahead of competitors results in products that are marketed as soon as possible, often without adequate testing, and ultimately in a succession of versions that requires the user to keep upgrading, migrating digital objects created with previous versions, learning new functions and commands, and so on. In addition, the software code is proprietary, so that any changes to it must be done at the source. Despite these obstacles, proprietary software is attractive since the competitive environment in which it is created encourages the development of products that will be more attractive, more effective, more efficient, and easier to use than the corresponding product of the competitor.

If we were forced to identify the single most problematic factor in using publicly-available tools in the classroom, we would say that it is insecurity on the part of the students, since all other difficulties, including technical difficulties, seem to be a

problem mainly because they cause insecurity among the students. In our two-year master's programme in information science, almost all students are registered in the professional stream. The courses offered are laboratory-intensive and there is very much emphasis on practical work and scheduled, supervised exercises that are usually completed in the computer laboratory. Usually the protocols for the exercises are very highly structured, spelling out procedures and results to be expected. The multimedia course has rarely used this format because of the creative nature of design of multimedia systems. For the practical work apart from the main project of building a prototype, students are given problems to solve, but little or no guidance as to what the solutions are or how to solve the problems is given. In addition, students are told that the laboratory staff cannot help them and that they are to find their own solutions. The objectives of each exercise are clearly stated in order to make clear to the students what is expected, but since they are used to working with structured, procedural protocols for the laboratory exercises, this approach makes them insecure. What does the professor want us to do exactly? What is the answer he is looking for? How can I know whether my response is correct or not?

In the first classes of the course, this less-structured approach is explained to the students, as is the necessity of such an approach in order to attain the goals of the course. Emphasis is placed on the need to rely on their creativity, on the fact that there are often several possible correct solutions to a problem and that they should strive to show that they have addressed the theoretical issues in the solutions they provide. They are assured that they will be treated fairly and they are encouraged to come up with solutions that are original. Despite the emphasis placed on these issues, many students quickly forget that they can use the course to express creativity, and they spend a good deal of their time worrying about what is expected of them and how they can arrive at correct answers when so little is known about the shape such answers can take.

Thus even without the use of publicly-available tools for building multimedia products, there is already some insecurity among the students of this particular course. The use of SVG, SMIL and so on adds to the insecurity because the software available for using these tools is relatively undeveloped, compared to many of the more stable and predictable software packages they use in other courses. The editor we chose to use for creating SVG objects (WebDraw, http://www.jasc.com/products/webdraw/) was available only in the Windows environment, but it offered a free 30-day trial period, which was enough for us to create an exercise during which the students could learn it and produce an SVG object as an exercise. The SVG object was an advertisement for the MetaMap, and resembling the animated GIFs seen all over the Web. In previous versions of the course the students had used proprietary software to produce animated GIFs as an exercise. WebDraw let the students create an advertisement using a graphic interface, but also permitted them to go in and edit the SVG code afterwards. Thus they could add comments explaining various components, remove unnecessary lines of code created by the software, and even add functions that came to mind after the digital object had been created.

We wanted the students to integrate some SMIL components into the Web site they were designing and constructed an exercise to help them do this. The exercise involved following a Web-based SMIL tutorial our research assistant had worked out

(http://www.esi.umontreal.ca/~moalv/blt6336/tutorielsmil.html), then creating a mini-production incorporating text, images, video, and sound, as well as the animated advertisement they had created in the previous exercise.

As with teaching the construction of Web sites, students were first required to work out the structure on paper. This involved:

1. Making a list of elements to include and identifying the corresponding computer files;
2. Drawing a representation of the timeline for the synchronisation of the elements, including the start and finish times of the elements;
3. Drawing a window to display the SMIL production;
4. Dividing the window into zones for displaying the elements.

The next step involved constructing the SMIL file, including information in the `<meta>` tag to identify the work, the `<root-layout>` tag for the SMIL window, the `<region>` tag for identifying zones in which the elements would appear, as well as the `<audio />`, `<video />`, `<img />`, `<text />`, and `<animation />` tags for the multimedia elements. Finally, the students were to use the `<seq>` and `<par>` tags to synchronise the elements.

The evaluation criteria for the exercise were: Conception, including imagination, creativity, and pertinence of the ideas (40%), Presentation, including correct synchronisation, readability, attention to detail (20%), and Contruction of a valid SMIL file (40%).

However, in testing the protocol for the exercise, and despite the many players available for displaying SMIL documents, we were unable to get it to work properly in the computer environment the students had available for their use. Ultimately, we decided to cancel this exercise. Some aspects can be explained but others remain a mystery. This is partly to do with our own lack of detailed technical knowledge of SMIL, and probably partly a function of the complex information we were trying to integrate. This was disappointing, since it was the exercise that showed the integration of all the elements in a Web-based environment, but since the term was ending and there was much pressure elsewhere, the students were glad. As a compromise, our research assistant did a demonstration of SMIL to show the workings and to offer at least some information.

Discussion

As we indicated earlier, there are two important problems in teaching multimedia using Web-based tools. First, the tools themselves are not yet mature and offer only a degree of stability. In addition, editing software that facilitates working with the tools is not as readily available in the marketplace as is software for other types of applications. It is true that people who develop expertise in reading and writing the code can use any text editor to create SVG and SMIL documents. However, once the graphics become the slightest bit complex, this solution cannot be considered practical or reasonable. The ideal set of skills to have is to be proficient at using an editor and to be able to read and edit the code it generates.

The second problem is the feeling of insecurity on the part of the students this approach creates. Over the years, we have observed that the best students do not

mind the ambiguities, perhaps because they tend to explore things further than average students do and are generally not so worried about the grades they obtain, concentrating rather on maximising their opportunities for acquiring knowledge while they are enrolled as students in the programme. Some of these are attracted to the ambiguous, less-structured approach to the practical exercises, which they see as a welcome change from the more structured, procedural exercises that leave no room for them to think and to create.

The feelings of frustration students often have are reflected in the evaluation for the course. The required evaluation for the courses at our school consists of a series of statements, in response to which the students indicate whether they are in total agreement or total disagreement or somewhere in between. The responses take the form of a numerical indicator and the results are reported as a statistical calculation. In addition, the students can add written comments, which are transcribed and printed before they are distributed to the professor, in order to ensure anonymity. Although the students rated the course well for aspects including whether the objectives were met, whether the course is generally well-structured, whether the exercises facilitate learning, and whether the professor knows the material and makes the link between theory and practice, some of the comments seem to contradict this. One student complained that the research assistant knew the software better than the professor did. Another thought the research assistant should give the course, since she was able to give better responses to their questions about the technical aspects. Two wanted more direction in creating the Web site. Another found that the instructions for the exercises were too vague.

We sympathise with these frustrations, but note that all these aspects were explained at the beginning of the course. We explained in the first class, while students still have time to change their mind and register for another course, that we were trying a new way to give the course, that the professor could not invest the time to learn the software and that the research assistant would be the resource person for the technical aspects of the course, that the state of the art is that things are not very stable, that there would be ambiguities, and so on, but that we felt it was worth living with these because they would have a richer learning experience and be better equipped for building multimedia products as these tools develop.
Our experience has been that often students do not understand the necessity of some of the required courses, the way some courses are taught, the content of other courses, and so on, and despite a concern for constantly improving teaching in our school, some of the complaints recur year after year. However, we also get feedback from graduates who gain an understanding of these things only once they have had a chance to work for a while and to put their new knowledge into practice. Only then does it become integrated and the flash of understanding achieved. We suspect that many more gain such understanding but do not come back and tell us. Thus we endure the complaints and inconsistencies in the student evaluations, and continue to try to innovate in our teaching methods, as long as we can believe that they are good ones.

Another important element of the discussion is the help that became available because of the presence of our research assistant. The demands on the faculty in our school are such that we could not have offered the course in this way without such help. By spending much of the summer in learning the tools and developing the

tutorials, our research assistant was able to develop proficiency in the use of the Web-based tools that was far beyond what faculty members are able to do. Indeed, this world moves so fast that we have observed that students in the second year of the two-year master's programme sometimes develop skills more advanced than those of the professors who taught them in the first year, partly because they have more time to explore the possibilites, and partly because they are often given the opportunity to integrate such learning into a project they undertake as part of the requirements for another course. Our general philosophy is that the university is a learning community, and the professors, the research assistants, and the students are all there to learn from each other. In the case of our multimedia course, the research assistant provided the link between the professor and the students which permitted the teaching of Web-based tools. Another element that was extremely helpful was that the research assistant had also been heavily involved in the development of the MetaMap, so she was very conversant with the technical issues.

In the fall of 2003 we will use the Web-based tools again in offering the course. We feel they are the way of the future and despite the difficult technical aspects, the advantages in providing such useful training for students in our programme outweigh the difficulties of delivering the course.

Conclusions

Teaching the use of Web-based tools that have not yet reached maturity involves some difficulties, but in deciding to take this approach we just have to live with them. Problems with the tools themselves have to do with the fact that although the tools have attained enough maturity to be released publicly, they are not yet completely stable. In addition, few editors or other software tools using the Web-based norms are available, as developers have little interest in investing in the production of such tools until the norm achieves a greater degree of maturity. Thus early adopters are required to be able to fend for themselves, must be able to fish around on the Web to find a community of users or other help created and published by other enthusiasts, and generally need to have some kind of faith that the problem is worth the investment of their time and energy. It is very much in the spirit of the Web community to adopt a collaborative approach, and to wish to make some kind of contribution to the overall picture in cooperation with others who share this interest.

Problems in delivering a course which makes the use of immature Web-based tools revolve mainly around student insecurities, but the instructor and assistants also face difficulties. It seems useful to engage the participation of students by emphasizing the positive aspects, such as their participation in innovation, being part of a community of testers, making a contribution to the development of standards for information management, and so on, in the broad context of organising the semantic Web. Our experience has taught us that it may help to remind them several times during the session of the positive aspects, rather than just laying out these ground rules at the beginning of the course, and we will do this next time the course is offered. The question of worrying about what grade they will get for the course should be de-emphasized, but this is not easy. From the point of view of the instructor and the teaching assistants, a considerable investment is also required in order to deliver the course adequately. First, much more preparation is required than that necessary for a course which involves proprietary software. This has to do with

the stability of the software and the predictability of outcomes and results of various interventions and manipulations. Not only the students but also the teachers must live with the ambiguities, the lack of adequate documentation, quirks in the software they are unable to explain, and unexpected malfunctions of the software. Emphasis needs to be placed on the evaluation students receive for the course, with many more points for participation and effort than for the results as such. This is difficult, since it does not correspond to the usual expectations students have. However, in adopting this approach we need to believe that the deeper knowledge obtained makes for a richer learning experience.

It is also clear to us that the help of a teaching assistant with a capacity for problem solving is critical to using Web-based tools to offer a multimedia course. In some schools obtaining this help is not a problem, and in others it is simply not possible because of budget restrictions. The point we wish to make here is that the heavy investment required in planning, preparing, delivering, and evaluating such a course means that it would be suicidal for faculty members who are already overworked to attempt to do this without the help of an assistant.

Despite our own enthusiasm for the two group projects that all students worked on in a single team when special circumstances arose, in both cases the students generally did not share this enthusiasm to the same degree. Thus we will return to the formula of having them choose their own projects for building a multimedia prototype, in the hope that they will have a better sense of ownership and be prepared to invest more in their coursework. Still, there are always a few students who have no idea what they might do as a project and who require an investment in discussing their interests, issues in their family or their social or economic environment, and so on, in order to tease out a project that will interest them.

Finally, the problem of the development of Web-based tools taking much more time than the development of proprietary software will not change any time soon. However, we need to emphasize that when the Web-based tools reach stability and some degree of maturity, they constitute much more useful end products, are publicly available, and assure longevity to the products created using them. Perhaps we can expect that in time, as the collaborative spirit spreads and as a much more ordered Web emerges, the time required for development of software tools will speed up, mechanisms for user testing will be streamlined, and that software will mature more rapidly. Meanwhile, in making an early transfer from research to the classroom, we will just have to live with the ambiguities, the instability, the dissatisfaction among some students, and the knowledge that by the time we deliver the course again next year, many changes to the software tools will probably already have taken place.