7. Acculturation and the Digital Humanities Community

Geoffrey Rockwell and Stéfan Sinclair

What graduate students want […] is simply answered at the present time: they want a job.

—John Guillory[1]

One can think through a digital humanities curriculum in three ways. One can ask what should be the intellectual content of a program and parse it up into courses; one can imagine the skills taught in a program and ensure that they are covered; or one can ensure that the acculturation and professionalization that takes place in the learning community is relevant to the students. This chapter will focus mainly on the third approach, but use that to touch on issues of content and skills.

Professionalization involves the development of skills, identities, norms and values associated with becoming part of a professional group.[2]

Acculturation, or as it is often called, professionalization, is the process of preparing students to fit into the culture of the professional community.[3] It is an often-neglected part of the curriculum that is tacked on at the end with a few workshops or an industry speaker's series. Good students should be able to figure out the professional culture by observing their supervisor or acquire it during the first months in a new job. Sometimes a gracious supervisor will mentor their students, but this mentoring is rarely planned.

Despite the historical lack of attention to acculturation, most undergraduate and graduate programs have begun to address it because of students' anxieties about getting a job after graduation, and because it is increasingly clear that the jobs available to them are outside academia. For these reasons we need to do more than model academic professionalization.[4]

How then is professionalization introduced into the humanities curriculum? Professionalization, and more generally acculturation, is—given its ambiguous place in the humanities—rarely introduced as explicit content. More often it is introduced through non-credit activities, reflecting a curricular bias toward concepts over applications:

1. Professionalization is woven into a course as a discussion topic in the final year like a Senior Thesis course.
2. It is delivered through workshops about specific issues.
3. Some universities provide internship opportunities that place students in professional contexts.
4. Professionals are invited to present their work and work culture.
5. University career centers often have general services including services to connect students to professionals for mentoring.

It should be noted, however, that the core curriculum does familiarize students to some professional activities—namely the narrow range of activities that are typical of academia. Professors in the humanities are expected to read widely in their field so we assign regular readings that familiarize the student to reading. Professors in the humanities are expected to publish monographs and journal articles (or perish) so we assign the writing of academic papers and theses. Professors are expected to give conference papers so we ask students to practice giving papers in seminars and student conferences. In short, professionalization happens even in the core of what we ask of students, but for only one type of profession—ours.

We basically prepare students at both the undergraduate and (even more so) at the graduate level to become like us and do what we do.[5] And that is the crux of the problem with this sort of curricular acculturation: it only prepares students for academic careers in their field of study, in which there are fewer and fewer jobs.[6] It is no surprise that, as the number of academic jobs dries up, people are asking whether it is worth doing a traditional doctorate aimed at an academic job or whether we should be preparing students exclusively for academic positions. As Peter Conn puts it,

> At a minimum, even if graduate faculty members themselves refuse to engage in training or advising students toward alternatives, they should destigmatize such decisions on the part of students and should support those who choose to explore

> careers outside the academy. Information about nonacademic careers should be included on placement websites. Among other outcomes, broadening postdoctoral career opportunities would serve the interest of departments eager to maintain higher rather than lower levels of graduate-student enrollments.[7]

This is why acculturation is so important and why it is important to think beyond academic professions. This is also one of the virtues of the digital humanities, as it is an intersectoral field that brings together researchers, librarians, computing staff and even industry practitioners.[8] In other words digital humanities is a field that is potentially broader than the academy. The jobs available to graduates already include non-academic jobs or *#alt-ac* (or alternative academic) jobs.[9] The question is how digital humanities programs can prepare students for a breadth of careers including academic careers. Having introduced the case for acculturation, now we will look at its purported goal—what are the objectives of digital humanities acculturation?

What Do We Do, Really?

One way to think about acculturation is to ask what professional digital humanists do and don't do and then ask how those activities are encouraged in the design of curriculum and community. It is useful to start with what digital humanists do not do, though we need to be clear that we are talking about what they don't *necessarily* do as part of being a digital humanist. We want to know what they do and don't do *qua* being a digital humanist. Some of the things digital humanists don't necessarily do are write books, teach credit courses and get academic jobs.

Don't Have to Write Books

Digital humanists don't necessarily write books. It isn't part of the job the way it is for English professors, even if many DH-ers do it. One of the ironies of graduate programs in digital humanities is that they are often modeled on traditional humanities programs that highly value the writing of sustained works of research like books, even though digital humanists typically don't write books and a number of digital humanists have received tenure without having written one. One might wonder if we should even insist on a written thesis or whether, like art programs, we shouldn't be open to capstone works in different media. Can one do sustained research that doesn't result in a book? Presumably the digital humanities is committed to the idea that digital work can be considered research.

Don’t Have to Theorize New Media

Though arguably regrettable, many digital humanists are not theorizing new media in recognizable ways. That is what new media studies, game studies, communications studies, philosophy of science and even English do well. The digital humanities is about developing things and it is experimental, though we would like to think that we experiment in a theoretically informed fashion.[10] The challenge in the digital humanities is to avoid a split between theory and practice and find ways that building can be theorizing and vice versa. Thus many are experimenting in the digital humanities with new ways to develop theory and therefore our work does not always resemble the theoretical outcomes of other traditions in the humanities. In other words we do not necessarily write theoretical works that encourage others to think through new media. Instead we try to develop new ways of engaging theory and practice. This is not to say that new media theory is unnecessary, it is just to say that many digital humanists are trying alternatives and therefore don’t have to be trained in traditional theoretical practices.

Don’t Have to Teach Credit Courses

Digital humanists do not necessarily teach university credit courses. Many digital humanists are not instructional staff and therefore are not required to teach as part of their professional responsibilities. That said, it is important to be able to explain technology and many digital humanists have to run workshops to train people whether in a library or for a project, but they don’t need to be trained to be undergraduate instructors (in fact, very few professors are formally trained to teach anyway.) There is a difference between the commitment to education of a professional pedagogue, which is what teaching professors should be, and the commitment of someone who occasionally needs to train others.

Don’t Have to Become Professors

In short, many digital humanists do not get faculty jobs, though they may work in learning institutions in other capacities. Many work in libraries or electronic text centers or computing services. This explains why much of the acculturation of graduate programs (and, to a lesser degree, undergraduate programs) are missing the mark.

So what do digital humanists do? The short answer is *projects in community*.[11] Digital humanists work on projects in interdisciplinary teams whose goal is to create rich digital works that make artistic, cultural and historical content available using computing. This is what acculturation in the digital humanities should prepare students to do. Meaningful experience working on digital projects is what makes digital humanities graduates (even when their degree is in another field, as most are) so attractive and this is what employers are looking for. This is professionalization that does not run the risk of being narrowly focused on a particular job category. This is professionalization that prepares students for academic and para-academic positions.

What then are the characteristics of digital projects for which we need to prepare students? Some of the things digital humanists typically do is work in interdisciplinary teams, apply digital practices to the humanities, manage projects or collaborate in the management, explain technology and build community.

Work in Interdisciplinary Teams

Typically digital humanists have to work with others in teams that will include content experts, librarians, computer scientists, software engineers, programmers, designers, videographers, GIS specialists and project managers. To thrive in projects students need to learn about these roles, the discourse of these roles and their traditions of formation. It is especially important to be able to work with people with a technical or scientific formation who may have little experience with the arts and humanities.

Apply Digital Practices

Creating digital works is a craft that requires the appropriate use of different media, technical skills, artistic and rhetorical competencies, and an awareness of the intended audiences and uses; each of these can be developed through a combination of teaching, apprenticeship and autonomous project work. For some kinds of computing practices, familiarity with development strategies is also useful, such as agile programming.[12] Digital humanists are typically expected to have a sense of the breadth of digital technologies and associated methods that can be applied to challenges in the humanities. They are at the interface between humanists and technologists advising on what should be done and then implementing technology-rich solutions.

Manage Projects

Digital humanities projects that are developed by a team over time must be managed. Students, therefore, should be introduced to project management. They should know how to use the tools of management and communication from wikis to issue tracking tools to conferencing tools.[13] More importantly, they should have been exposed to project management strategies and the discourse around project management, so that they can fit into teams and think critically about how projects are managed and about their role in its management. This is not to say that all projects need explicit management, just that students should be prepared for complex projects in which they have to reflect on management.

Explain Technology

While we argued above that digital humanities students will not necessarily have to teach university credit courses on information technology, they probably will have to explain technology to people—from team members to potential users. DH students often have to write documentation, train people to use a system, track bugs, interface between content specialists and programmers and give public presentations about projects. They should therefore understand the technologies they use critically, learn to use technical language accurately, be able to patiently explain technical concepts to those who do not have a digital humanities background and be able to present technical projects effectively.

One could argue that we are brazenly trying to define the discipline in this discussion of what digital humanists do; that is not the case, at least in the traditional sense of defining a canon. We are arguing that the training students receive should reflect what they will have to do as professionals, instead of unthinkingly mimicking traditional graduate training in the humanities. A new field in the early stages of developing its academic practices should take advantage of the opportunity to question graduate training and experiment with alternative models. To the extent that disciplines are about self-replication, (the formation of disciple students) we admit to trying to model the discipline. We propose that we should aim to train students to be able to participate as professionals rather than grafting old habits onto the digital. To do this we have to think about how we plan, maintain and audit programs, which is what the rest of this chapter will focus on—ways of thinking through programs and the acculturation they facilitate.

Case Study 1: Multimedia at McMaster University

At this point we will illustrate how acculturation can be woven holistically into curriculum at both the undergraduate and graduate level with two case studies. The first case is the undergraduate multimedia program at McMaster University and the second is the graduate MA program at the University of Alberta. Both authors have worked at both institutions and have been involved with the design and delivery of both cases.[14]

The undergraduate program in multimedia at McMaster was developed in 1998 in response to a call from the Province of Ontario for expanded programs that would prepare students for careers in advanced technology. Rockwell led the preparation of a proposal to the Province from the Faculty of Humanities for a Combined Honors in Multimedia (and another subject).[15] The proposal, which was funded, built on humanities computing courses that Rockwell had developed from 1995.[16] Thanks to the generous new base funding that came with Province of Ontario's Access to Opportunities Program (ATOP), the program was not developed out of cross-listed courses across the humanities as many programs are. (We recognize that this was a luxury not shared by all as administrations are pressed to invest more meaningfully in new digital humanities programs without having the base funding to create a program from all new courses). This allowed us to develop the project logically, to build proper multimedia facilities for the students, to hire faculty *specifically* for this program and to have a core of hands-on studio courses that are taught in sections small enough for students to be able to learn skills in the context of meaningful assignments.

Rockwell designed the multimedia program based on the definition of a multimedia work: "A computer-based rhetorical artifact in which multiple media are integrated into an interactive artistic whole."[17] There was thus a core of mandatory courses that dealt with the creation of multimedia works through "integration" of multiple media and then courses on individual digital media like electronic texts, digital images, digital video, animation, electronic music and so on. Since then courses on subjects like Web Programming have been introduced, while others like Writing in the Electronic Age have been dropped, but the logic of a studio program that focuses on the creation of multimedia works remains.

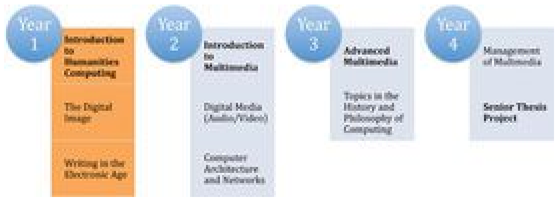

Year 1
Introduction to Humanities Computing
The Digital Image
Writing in the Electronic Age
Year 2
Introduction to Multimedia
Digital Media (Audio/Video)
Computer Architecture and Networks
Year 3
Advanced Multimedia
Topics in the History and Philosophy of Computing
Year 4
Management of Multimedia
Senior Thesis Project

Figure 1. Core courses in the multimedia program at McMaster University.

This logic can be seen in the mandatory core courses. In the chart of these courses (Figure 1), the bolded courses have assignments where students create projects that integrate multiple media. The others deal with individual media or some important related topic. To get into the multimedia program, students had to take two of the three first-year courses and show facility with the creation of multimedia. (Entry into the program is limited to cohorts of the best first-year applicants, as we have limited space in studio classes and limited space in the labs.) These first year courses also serve as general digital arts and humanities courses for students across the faculty—indeed, they have proven to be very popular with senior students looking for an elective that prepares them digitally, which has required us to develop strategies to ensure that prospective program students aren't denied seats in the first year courses.

The Introduction to Humanities Computing (now Introduction to Digital Media), for example, is a course that can handle hundreds of students with a large lecture every week and a tutorial in smaller groups of twenty in a lab where students build website essays. The tutorial labs are taught primarily by fourth-year multimedia students as well as, more recently, some graduate students with a multimedia background, which gives them paid work experience explaining the skills they have learned, in addition to reinforcing or expanding basic skills. (It's amazing how easy it is to forget how to create a sophisticated web page, but technologies—like CSS—evolve so quickly, that refreshing one's technical competencies is valuable.) In our experience, being a teaching assistant in their fourth year is a great way to take responsibility for learning and managing learning. This paid work experience professionalizes the fourth years by preparing them to be able to talk about technology and train people.

Given the focus on the creation of multimedia at McMaster, it will not come as a surprise that there is a strong project focus to the program, with most classes culminating in some sort of collaborative project.[18] While project skills are developed throughout the program, project work is dealt with explicitly in the capstone experience in the fourth year where students must take a Management of Multimedia course in the fall term followed by a Senior Thesis Project course in the winter term. These two courses are coordinated. In Management of Multimedia, they plan their thesis project dealing with project management issues like intellectual property, budgets, timelines, writing proposals, presenting ideas, building teams, working with clients, working in teams and documenting projects. The Senior Thesis Project is run differently, as students are let loose to do the project with minimal supervision and no required class meetings. While Management of Multimedia is a class with weekly meetings and the usual structure of assignments, the Senior Thesis Project is run like an independent study or thesis project. Each of the supervising professors has (credited) hours set aside to meet the project individuals or teams, but the students have to choose to come in and consult with their supervisor. The idea is to encourage students at the end of the program to first plan their project and then to actually manage the planned project through to implementation without the usual academic pacing of class meetings. The effect for most students is empowering, though a few get lost and leave the completing of the thesis until the last moment with predictable results. All this culminates in three or four days of presentations open to all, which highlight and celebrate the creativity of the cohort.

Competencies and Curriculum

While a definition of multimedia may have provided the logic for the types and deployment of courses, we also used another technique to help us design and then audit the program—to think through the *competencies* we wanted to teach. In the curricular literature, "competencies" are usually associated with vocational skills. The idea is that when designing a curriculum you start by identifying what you want graduating students to be competent at. This encourages curricular designers to describe in concrete terms what graduates should be capable of doing so you know that they had the competency. Competencies, in curricular planning, have the following features:

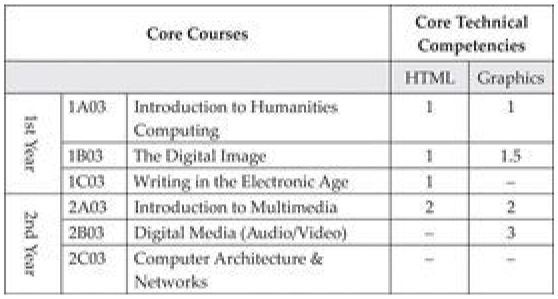
- They are used to describe *what students can do*, not what you are going to teach, thus focusing planning on student achievement. It is too easy to plan curriculum in terms of abstract subject components.
- They can be used to *complement other ways of planning* curriculum. Most of us begin with a list of courses that cover the important content. Competencies allow you to track important components that cross courses—especially skills that have to be carefully staged and reinforced across multiple courses.
- They should describe things that students can demonstrate. This makes it easier to imagine assessment techniques. When describing competencies the idea is to *describe outcomes as behaviors* that would indicate successful learning.[19] Once you have a description of student successful behavior you can imagine authentic assessment activities. This then connects the planning of curriculum to the planning of individual courses. In effect, you can reverse engineer individual course outcomes from what you want the successful graduating student to be able to do. Teaching is thus not an end in itself, but the creating of a context for student achievement.
- They should be *comprehensible to students* so that they can take the measure of what they are learning and compare the curriculum to their expectations. If they are articulated in accessible language then they can be used to communicate what the program is about to prospective students and to incoming program students. Ideally students should get to the point of being able to negotiate (and critique) their

education in terms of the competencies developed. They should be allowed to contribute to the ongoing curricular planning and auditing; competencies enable them to think about their education. It allows them to say to you, “I thought I would be able to do X by the end.” Competency planning, if negotiated with students, also gives them tools for reflecting and managing their own learning after graduation, a general goal of student-centered education and an important competency in itself; students are in a better position to articulate their strengths to prospective employers.

- Many government, educational and industry organizations describe the *competencies expected of people in industry* and the world, which allows program participants, both students and instructional staff, to compare how students are prepared to what is expected or recommended by practitioners.[20] This is not to say that curriculum should be designed to fit industry or government mandates, it is to say that if university programs are to prepare students for a breadth of careers you need to have a vocabulary for comparing student preparation to expectations in professions. Competencies are one way to do that.
- *Job ads are often articulated in terms of competencies* desired in applicants. Students and instructors who are comfortable discussing competencies thus have a way of assessing job prospects and suitability of different types of jobs. Whether we like it or not, students are justifiably concerned at the end of their program with their career options and job prospects. To wave our hands and try the old platitudes about the value of a liberal arts education doesn’t help them with the real stress of finding their way after university. Without narrowcasting a program for specific job types, the discussion of competencies can give students a way of thinking about what they can do.

In our case, we didn’t think of competencies in the limited sense of technical skills, but we used it as a general rubric to gather all the knowledge, skills and abilities that we felt were important. This allowed us to develop one rubric of what we wanted students to know, understand and be able to do technically in one place. This included what we called intellectual competencies like “know about the history of computing so as to be able to discuss the history of computing and better understand the contemporary context” with technical competencies like “have the skills to be able to create sophisticated web pages” and social competencies like “be able to play both leadership and contributor roles in groups.” The list of competencies were then negotiated, simplified and grouped into *core technical* competencies (that all graduating students should have), *core intellectual* and *others* which included social competencies and elective competencies (that students should have if they wanted and took the requisite course). For a list of the competencies we were using, see Appendix 1.

Finally these competencies were mapped against the courses (see Table 1). We started the mapping with the assumption that any competency that we care about should be introduced early on and then explicitly developed in a second or third year course (by which we mean that the competency should be taught in a module and there should be an assignment aimed at testing competency), and subsequently reinforced in one or more additional courses. To test this we assigned to a course a [1] for competencies introduced, a [2] for those explicitly taught, and [3] for those reinforced. These assignments were placed in a spreadsheet with all our courses down the side and all the competencies across the top. The results were instructive. Some competencies were often introduced but were never dealt with explicitly at any length. Some competencies were not reinforced. Some courses were overloaded with [2]s so that they had to carry too much of the load. And some competencies that we thought were important were only dealt with in elective courses. This gave us a way of balancing the curriculum and arguing for resources from the administration.



Core Courses			Core Technical Competencies	
			HTML	Graphics
1st Year	1A03	Introduction to Humanities Computing	1	1
	1B03	The Digital Image	1	1.5
	1C03	Writing in the Electronic Age	1	–
2nd Year	2A03	Introduction to Multimedia	2	2
	2B03	Digital Media (Audio/Video)	–	3
	2C03	Computer Architecture & Networks	–	–

Table 1. Portion of an early competency chart.

Not only were the competencies used to design the program; they were also used to audit the program as we changed courses (this is akin to unit tests in programming).[21] They also helped us introduce new instructors to the way courses they were hired to teach fit in the curriculum. It allowed us to show sessional instructors what competencies they were responsible for introducing, teaching explicitly or reinforcing. This becomes especially important in the case of technical skills that build on each other. A student can be expected to use digital video in a third year course if the instructor of the second year Digital Media course has taught the expected competencies. Last of all, and to return to the subject of this chapter, this competency approach let us plan a curriculum in an integrated fashion, during which we took into account content (intellectual competencies), technical skills and acculturation in one planning/auditing matrix.

Case Study 2: MA in Humanities Computing at the University of Alberta

The MA in Humanities Computing was launched in September 2001, following two to three years of planning and development.[22] In a similar fashion to the multimedia program at McMaster, a one-time funding opportunity (ACCESS, a government of Alberta funding agency) had a catalyzing effect: the additional funding allowed for a proper investment in infrastructure and personnel, but also meant that funds were not being begrudgingly siphoned from other areas. Susan Hockey and Patricia Clements led the initial planning, though consultation was broad across the Faculty of Arts, the University of Alberta and beyond (Ian Lancashire and Harold Short were consulted, for example.) It is worth noting that several graduate students contributed significantly to the initial planning of the program—this involvement is also a form of academic acculturation.

The MA in Humanities Computing, which was created before the term "digital humanities" was coined and widely adopted,[23] confronted many of the same challenges that similar programs might face today (though at the time with few precedents from which to learn). For instance, we recognized that the MA program would include students from across the humanities, social sciences and fine arts, so we wanted to ensure a broad representation of intellectual perspectives, while also allowing students to develop deeper expertise in specific domains. By and large, this balance was accomplished by incorporating a wide range of issues into the seminar discussions and allowing students to create more narrowly focused projects. This transfer and adaptation of skills from the general to the specific is clearly relevant for a range of professional contexts.

Another common challenge in developing a digital humanities curriculum is in striking an appropriate balance between theoretical and practical components.[24] We strive to train students to get up to speed as quickly as possible with technical skills that will allow them to engage effectively with tools, and—in some cases—adapt those tools or even create new ones. At the same time, we are wary of limiting ourselves to specific software packages and methodologies. For instance, we may want students to understand the fundamental techniques and broader implications of digital image manipulation; this could involve becoming familiar with Adobe Photoshop, even though competency in Photoshop would be a secondary objective. Likewise, we encourage students to understand and critique the limitations of existing text analysis tools;[25] this can help them to recognize how a small amount of programming knowledge can empower them to accomplish idiosyncratic but extremely useful tasks.[26]

The Humanities Computing MA is a two-year program that ends with a thesis as a capstone experience. Over the two years, students take four mandatory humanities computing courses and five electives, two of which have to be humanities computing courses. Students can choose to study humanities Computing alone or do a specialization in another subject like English, philosophy, art and design, or in any of the Faculty of Arts departments that offer a specialization with humanities computing. This interdepartmental option gives students the ability to apply digital humanities practices to problems in their specialization so that they could continue on to a PhD program in their area of specialization. If they choose a specialization, they then have to apply to both units (i.e., humanities computing and the department of specialization), take at least two of their electives in the area of specialization, and have their thesis co-supervised by someone from each unit. In addition, we have a special three-year joint MA/MLIS that gives students both the professional library and information science degree and the academic arts degree.

Ultimately, the two-year timeframe of the MA in Humanities Computing is extremely short for building a foundation in digital humanities, while also providing core skills training. This is especially true since humanities computing is typically an entirely new discipline for incoming students—it is not, as with most other graduate offerings, an extension of an undergraduate degree. We have explored several solutions to this dilemma, including a skills boot camp when students first arrive, as well as offering a separate stream of workshops (intended for first year students and offered by second year students). One of the dominant guiding principles for us has been to try to anticipate what balance of theoretical and practical skills would be most valuable to students after they leave the program, whether they stay in academia or not.[27] Currently, the program has the following two formal components that help acculturate new students to the digital humanities.

Intensity Experience

In 2009 we introduced an “intensity” experience at the start of the academic year. The first week of classes is cancelled and all new students are formed into teams with returning students and students from other programs that volunteer. These teams are given a week to make progress on a challenge that is typical of a professional project. For the first two iterations the challenge was to develop an Augmented Reality Game (ARG). The teams are given a tour of the resources that are available (from labs to the library) and let loose to figure out how they would develop an ARG that, for example, can be used for health education. They are expected to present what they developed a week later. Needless to say, they do not complete the project, but in the spirit of problem-based learning the idea is that they should start their program with a real challenge; typical of what they should be able to do by the end of the program.[28]

We call this an "intensity" experience because the idea is to ask students to focus intensely on one challenge at the beginning, rather than dividing their time among different classes. This is partly designed to orient students to the field by giving them an initial experience that is more typical of the field, rather than giving them more classes like those they had as undergraduates. We ask them to work in a team on a project that will require different skills, many of which they do not possess: to think about how they organize their team, how they break the project down, what they can achieve and how to present it back. While the intensity experience is a non-credit activity, they respond enthusiastically. In informal conversations afterwards, we have been told that the intensity experience dramatically changed students' views of what they were getting into—for the better.

Technical Approaches and Project Management

Our four mandatory first year courses can be broken into two streams with a fall and winter course. The first stream is composed of more traditional courses, namely, Survey of Humanities Computing in the fall followed by Theoretical Issues in Humanities Computing in the winter. The second stream includes Technical Concepts and Approaches in the fall and Project Management and Design in the winter term. Students are first introduced to the encoding and programming that they need to build a database-driven website in these courses, before practicing their technical skills in the context of a community project. Technical Concepts and Approaches covers HTML and CSS (with attention to XML), PHP (as a first programming language) and mySQL (as an introduction to databases.) In Project Management and Design, students are divided into teams of two to work on a project organized by the University of Alberta's Community Service-Learning (CSL) unit (http://www.csl.ualberta.ca/).[29] CSL works with community organizations that want digital projects done. Proposals are presented to the students who then complete them over the term. The projects can range from developing a database for materials in an activist organization office to a needs-analysis for a website for an organization. These projects rarely call on all the skills taught in Technical Concepts and Approaches; in fact, the projects often call for skills that the students did not learn in the course, such as when a community organization wants a new component to a website already developed in another framework. Nonetheless, the course gives them a real-world experience in which they have to adapt the skills they learned to a real need for an organization outside of the university. As part of completing the project they have to:

1. Identify and negotiate what can be done in the time available;
2. Agree on tasks and a timeline;
3. Work with the client organization learning to communicate effectively;
4. Develop a digital deliverable, document it and deliver it so that it can be used; and,
5. Finish a project, present it and identify next steps.

Class time in this course is shared between working on the projects and discussing challenges. While any one team may not have the skills needed for the project they have undertaken, there are almost always other students with ideas or relevant experience. This way, all sorts of techniques are introduced as needed by teams. If a team needs to develop an interface to a web tool, for example, this is brought to the class for discussion.

Apprenticeship through Research Projects

There is, however, another less formal way that students at the University of Alberta are brought into the field of digital humanities and that is through their research assistantships. These research assistantships are the primary way that we currently fund graduate students, as we don't have an associated undergraduate program that needs teaching assistants. While these research assistantships are limited by funding, we have been able to offer them to all incoming students in recent years, though not all students take advantage of them because of other commitments.[30] Research assistantships have the advantage that they involve students in real research projects that call for the skills they are learning, such as digital research and development.[31] In the Project Management and Design course, they are applying their learning to projects that typically are not academic and which do not involve research in the digital humanities. In research assistantships, students apprentice in teams using digital research practices.

The challenge, as with all research assistantships, is how to involve students new to the field in a meaningful way, giving them tasks they can accomplish when they are still learning the skills needed by the project. As anyone who has employed a new MA student as a research assistant knows, there can be a conflict between the needs of the project and the skills of the assistant. Research projects often have to develop digital solutions quickly (e.g. in a year) so that researchers can use them. Managers of such projects often don't want to take on a student who, having just started, has to learn on the job. Pragmatically, it is often cheaper and faster to hire a programmer, experienced encoder or trained graphic designer. We have addressed this tension by assigning assistantships strategically—so that no project has only new research assistants—and by supporting colleagues taking on MA students new to the digital humanities. We have developed a rough series of tasks that incoming students can tackle, as both a benefit to the project and as a way of learning professional skills. If you pace students properly and challenge them incrementally, they can learn the skills they need in order to participate, while helping the project concretely. This pacing of tasks is, in effect, an apprenticeship in which students learn through a series of gradually more complex challenges. The series of challenges involves the following tasks, though the order varies from project to project: an environmental scan, maintaining an open research site, a literature review, reporting to the team, interface design, and preparing conference proposals and papers.

Environmental Scan

Students new to a project are asked to conduct a scan of all the projects in the environment that have similar goals to theirs. They typically start with the grant proposal, which will have identified similar projects, and are then asked to broaden the list and to document what they find so that they can present it back to the team.

Maintain an Open Research Site

Most of our projects follow an open research model in which all of our documents are shared both within the team and with anyone interested through wikis, blogs and other project management tools like Basecamp (http://basecamphq.com/). An incoming research assistant can be asked to take over the management and updating of the project's documentation. To do this, they will have to learn whichever collaborative documentation tool is being used and they will have to read what is already there—both useful activities. We find the difficult part is training the graduate student to maintain the site; this is about habits of careful documentation of activity. Every time a new project is found that relates to our project a useful summary with links should be added to the environmental scan wiki page.

Literature Review

A task that someone with a good humanities degree should be prepared to do it is review the relevant literature. Incoming students should have the skills to find relevant literature and summarize it. By and large, incoming students are able to do this, which is why it makes a useful sub-project for them at the start of their research assistantship. We generally ask them first to find reports, articles and other documents reporting on the projects found in the environmental scan, and then to move on to the more open-ended task of searching the broader literature for interesting materials. Often students find the open-ended search daunting, as they don't know what others in the team might find interesting. A tactic we use is to have students first gather lists of anything that might be relevant without reading the items. We then go over the list, discussing what we are looking for, eliminating items, and prioritizing items that seem promising. We then ask the students to skim the items and report back to the team. Students are asked to identify the key ideas gleaned and arguments put forth, which leads to a further prioritization. In the last pass, students are asked to read the items prioritized carefully, to write a précis for each, and to create a bibliographic entry in our shared bibliography with the précis. The shared bibliography is itself another form of documentation that they take responsibility for. Nowadays we tend to use Zotero (http://zotero.org/) and form a Zotero Group (http://zotero.org/groups/) for each project. Students thus have to learn, in this context, another technology that is useful in the field.

Report to the Team

An important activity associated with both the environmental scan and the literature review is ongoing reporting. We tend to have weekly "lab" meetings in a space where we can project from our laptops. At these meetings, students report back their progress, issues are discussed (such as exactly what are we looking for in the literature?) and research assistants make presentations. It is tempting to save time and not meet until there is some final outcome, but it is in these weekly meetings that a lot of informal acculturation and guidance takes place. Meetings pace students and keep them on track. Most importantly, students learn when you care about what they are doing. They can tell if the research assistantship is a make-work project because in these cases, one is not interested in the results. We tend to ask students to report often, and use the reports to train them so that we can trust them to do vital tasks like the environmental scan and the literature review. For this reason, at weekly labs we give immediate feedback when the tasks are not done in a timely fashion or when they aren't done in a way that the team can trust to use in their research. Nothing motivates a student to be careful with details in compiling a bibliography more than when we find obvious errors of transcription in the report during our discussion of it.[32]

Reporting to the team also builds student confidence in presenting research and the discourse of research. By the time students are asked to present at conferences they should have presented short reports and drafts of things over and over to a friendly (but critical) team. Finally, the usefulness of reporting lies in what we learn from the students. At some magical point in the year, we find ourselves learning something new and exciting from the students. When they get to the point of bringing these contributions to us and motivating others is the time when you enjoy the benefits of the apprenticeship—when a research assistant contributes rather than distracts from research. Lab meetings that are completely sidetracked by some original idea from the research assistant are, we believe, one of the rewards of research—to learn something new in community.

Interface Design

So far the tasks we have described are not central to the creation of digital works. Because many students don't have programming skills, one way to involve them in development is to have them manage the interface design of a digital work. We favor a Personas/Scenarios/Wireframes/Designs approach, in which the student is first asked to develop "personas" or archetypal users and then generate usage scenarios that can be used to plan the software and audit the interface.[33] From the scenarios they can develop wireframes that communicate the functionality of the system being developed, and then produce graphic designs for the finished interface. They can then work closely with the programmer to implement the system testing the iterations against the prioritized scenarios. Finally, they can write the "about" text for the system, the online help text and user manuals, as well as other forms of documentation.

The point of this approach is that research assistants are given responsibility for a staged process—they are not being asked to go away and come back with a polished design. At each stage they have to work with the stakeholders to develop a consensus. Thus, when developing personas, they might start by interviewing stakeholders and potential users. From the interviews they might develop five possible personas and return them to the research team and content experts for refinement and prioritization. This managing of a negotiation is important for projects in which the digital humanities group is developing a digital presence for a larger research team. Negotiating personas is a way of making sure the digital humanities group is serving the larger research needs of, and is in communication with, the larger group. The research assistant becomes a vital conduit for this communication. Further, the research assistant serves as a vital communications link to stakeholders outside the digital humanities team.

One advantage of this approach is that it also stages the learning the research assistant has to do in order to usefully lead the process. They don't really have to learn any technologies until creating wireframes, and there are easy-to-learn tools for that.[34] The trickiest part is if you ask the research assistant to go from functional wireframes (which just shows functionality with little color, font, layout and art) to actual graphic designs. In a wireframe you don't want any touchy design ideas creeping in, as they tend to distract stakeholders with strong views about design.[35] A crude wireframe that is well annotated is an advantage. The moment you move to graphic designs the research assistant has to have a sense of web design. In many projects, this is where a professional designer might be brought in to create Photoshop mockups that the research assistant can use for negotiating final designs. Where a research assistant can learn is in translating the mockups into HTML and graphics for the programmer. This presents an opportunity to deepen their knowledge of HTML and CSS, and learn how to create web-efficient logos using graphics tools like Photoshop.

Conference Proposals and Papers

A last task that inexperienced research assistants can help with is developing conference proposals, writing draft conference papers/slides and delivering the papers. This process starts in the fall, when the calls-for-papers come out. In the group we will discuss the calls and what we will have to present. A research assistant will be assigned to write a one-page draft proposal with an outline of what will be presented. We go over this in the lab meetings to refine it. We often find that we have to rewrite the opening paragraphs, as conference proposals are a genre of writing that students are not typically familiar with. Iteratively discussing a proposal and occasionally rewriting it is a very different way of training students to write for the field than assigning papers and marking them. They know the proposal will not go out until it is acceptable to the team, so there is more at stake in the writing. They are also told that if the proposal is accepted they will get to write the full paper and deliver it, which means a trip to a conference (for which we budget in grant proposals). This also provides a context for discussing research credit.

If the proposal is accepted, research assistants then work on drafts of the paper. Even if it isn't accepted by a national or international conference, we will ask the research assistants to draft full papers for local presentation (partly so that we have drafts of papers for the next round.) They have already done much of the legwork putting together the environmental scan and literature review. Often we first ask them to put together slides and present the paper informally. Only once we have refined what will be presented do they write a draft, which again they present in a lab. This helps train them to read papers and gives the team a chance to give detailed feedback. We usually end up editing the final version of the paper and the research assistant presents it. If the conference paper is of a high enough standard, we then discuss how it should be edited for submission to a journal—but that usually takes place in the second year of their research assistantship as the conferences are mostly in the early summer.

We cannot overemphasize the amount of learning or the sense of accomplishment that comes from managing the interface design of a system that, by the end of the academic year, is working and has real users. Likewise, we can't overemphasize the sense of participating in the larger research community that comes from presenting at a national (that is, Canadian) conference like that of the Canadian Society for Digital Humanities / *Société canadienne des humanités numériques* (CSDH/SCHN). Students are motivated by the opportunity to actually contribute publically to the field and learn by doing for others. The acculturation of this sort of apprenticeship is not a simulation of professional activity; when they present and take questions from active professionals at a conference, they are no longer acting at being a professional the way they might do in the simulacra of a seminar—they are being professionals, thereby acquiring the confidence that comes from presenting research to our peers.

Some of the sense of accomplishment comes from learning on their own, as they need to in order to accomplish useful research tasks. When we ask a student to create wireframes for a design we don't give them much guidance and we don't teach them any tools. The student has to figure out what a wireframe is supposed to do for us, figure out what tool they want to use and figure out how to present it back to us. Often students come to the next lab meeting lost or with inappropriate reports. They then have to learn to ask for help, where to get help and how to take criticism. When we rewrite a conference proposal from scratch because it was entirely unsuited for the venue, there is an implied criticism that they have to handle and learn from if they want to be tasked with the next proposal. We can characterize this apprenticeship approach as a reversal of the way students are taught in training courses, in which they are usually trained in a tool and shown successful models before they try their hand. In an apprenticeship, they are asked to do something the way a professional would be asked, and then helped as needed. It is a form of learning to swim by being thrown into the pool—or at least being asked to jump into the shallow end. When, after a year of successively more complex tasks, they see that they have contributed to real outcomes they realize that they have learned *how to achieve such outcomes on their own*, or at least *how to learn how to achieve on their own*. This is what training professionals in the digital humanities is about—preparing students to be able to do digital research on their own, where they continually have to learn new skills as needs be.

Needless to say, this is an ideal set of apprenticeship tasks. The tasks are not sequential (i.e., interface design has to start immediately and the literature review is iterative.) No grant-funded projects need the same tasks, and many projects that students are brought into are ongoing, so that some of these tasks were completed before. Each graduate student research assistant is also an individual, with his or her own strengths, weaknesses, and desires. We try to match research assistants with tasks for which they are prepared, tasks they want to do, or tasks that will help them with their own research, but often the tight timelines of grants mean that they have to do tasks they are unprepared are. We try to stagger the tasks so that they can learn incrementally, but sometimes the project needs things done in a different order. Sometimes the timelines of the grant conflict with the academic schedule of a student and they are afraid to say “no” to a request for a report just when a paper is due. Some students do such good work they get loaded with more and more tasks until they work beyond the hours contracted whereas others seem to never finish anything so they get marginalized in a project that has to meet deadlines. Some students thrive on the stress of learning new things quickly and seeing their work implemented, while others get anxious and are reluctant to ask for help because they see successful peers thriving and worry that they should know basic things. Caring supervisors and students thinking about such relationships can manage these difficulties. Supervising graduate students is a human art, which all thesis supervisors should learn and graduate students should consider. It is an art on which much has been written and at which we get better with experience.[36] It is beyond the scope of this chapter to discuss how to handle these inevitable tensions other than to say that more experienced research assistants can help. We are, however, convinced that students who apprentice in the field are better prepared, feel better integrated into the profession and are more likely to complete their program.

In conclusion, with regard to growth, graduate education appears now to be a kind of pyramid scheme. The prospect of its collapse has revealed something extraordinary, that the growth of literary study consists largely in the growth of graduate programs and in the transformation of graduate students into a public for literary criticism. Professors of literature now write and teach for graduate students; graduate students have become their constituency and collectively now exert a considerable pressure on the profession, moving it in certain directions, along the cutting edge of criticism. Hence the most symptomatic professional desire one can harbor today is expressed in the desire to teach graduate students in preference to undergraduates.[37]

A critic might object to the focus on acculturation in this chapter as premature. Faculty members hired before there was a job crisis may remember being able to concentrate only on their intellectual development without having to worry about learning how to teach, how to present conference papers, or how to now manage the design of a research interface. There is a feeling that we should create a safe space free of professional and cultural concerns in which students can develop their research thinking and that acculturation should take place only after intellectual development. Such views assume one can isolate the intellectual from the cultural. Jennifer Wicke describes "an institutionalized reluctance to admit that undertaking a PhD in the field [that is, English] constitutes entering a professional arena with rules, guidelines and protocols that may remain unarticulated."[38] The Modern Languages Association Ad Hoc Committee on the Professionalization of PhDs summarized the debate and made some safe recommendations in their 2002 report, *Professionalization in Perspective*.[39] They asked, "Given the intensity and elaborateness of the public discussions on this topic in recent years, how can we explain the continuing resistance to professionalization in general and to professional training of graduate students in particular that we have witnessed so often in our consultations?" They concluded that departments have to take responsibility for "the difficulties faced by the graduates they produce" and they should at the very least provide career counseling comparable to what undergraduates get.

In the paper quoted at the beginning of both this chapter and the conclusion, however, John Guillory makes a more nuanced point about the dangers of professionalization. He argues that professionalization can encode in a program the desires and politics of the stakeholders. He ends by accusing the system of becoming a pyramid scheme, in which the professional desire is to do research and teach graduate students, so that is what graduate students are trained to do, which then creates the need for more graduate students than are needed as professionals.

It would be easy to say that the approaches outlined in this chapter avoid the recursive danger of such a pyramid scheme. We could argue that the acculturation approaches discussed here aim not at producing more students who only want to become professors like us. We could and have argued that the digital humanities has an opportunity to prepare students for a breadth of careers, thereby avoiding the worst of a doctoral job crisis where there are too many graduates prepared only for the few new positions—and we would be right. But that shouldn't blind us to the danger of self-reproduction inherent in professional preparation. When we design programs, even ones that are designed to prepare for a breadth of careers, we are designing for the careers we know and the types of jobs we value or desire. In a fast moving field like the digital humanities, such preparation is usually tailored to what is past or current, not what may be in the future. Students who apprentice on text encoding projects will know little about areas like crowd-sourcing, serious games and physical computing. We can only hope that students begin to anticipate what will be innovative when they are looking for professional positions; that they will have learned how to learn new technologies and that they may even redirect the field with their enthusiasm and fresh views. In Appendix 2, we offer some advice and linked resources for students who want to think about professionalization and for curricular developers wishing to integrate professionalization into their digital humanities curriculum.

The paradox of designing programs that replicate our ideas about the profession is that most of us now designing the programs were not formally prepared in digital graduate programs. Most of those senior enough to design programs were trained in the digital humanities outside the programs they were awarded degrees in, and they were trained often in the face of discouragement. This has led to a romantic view of acculturation as a heroic overcoming of tradition-bound disciplinary values. As much as it may motivate students to think they are joining the revolution, as humanists we should be skeptical of our own fantasies of creation. Vico concluded that institutions were born in crimes (against the institutions they pushed out of the way.) What crime was committed against whom in the creation of graduate programs in digital humanities? What are we losing as we redesign the graduate experience? Are we perhaps encoding into the new culture our desires for revenge on the disciplines that couldn't fit us? One ideal of the humanities that we should remain committed to is that of self-reflection or knowing ourselves, not only individually, but also professionally. As Guillory concludes,

> What I call preprofessionalism is nothing other than the realm in which the profession's fantasies, both professional and political, are acted out. The kind of sociological analysis I have in mind will demand that we suspend some of our investments in specific agendas of professionalization and politicization in order to clarify what is merely phantasmic in those investments. The decline of the job market is a reality check, then, and perhaps an opportunity.[40]

Appendix 1: Multimedia Competencies (2001)

Core Technical Competencies

- Build a sophisticated WWW site
- Create bit-map and vector graphics
- Create an interactive CD-ROM
- Create time-dependent media (audio and video)
- Set up and network a PC or Mac
- Use a WWW server
- Create interactive works

Elective Technical Competencies

- Create and study an electronic text
- Create electronic music and compose on the computer
- Design a typeface and design a publication
- Create an animation
- Create a virtual space
- Create instructional materials
- Create an electronic presentation

Core Academic Competencies

- Be able to discuss the design of a multimedia work
- Be able create rhetorically effective multimedia works
- Be able to read critically, write effectively, analyze problems and solve them
- Be conversant with the history of multimedia design and information technology
- Be aware of the social, political and ethical issues related to multimedia technology

Other Competencies

- Be able to work in groups and understand the management of multimedia projects
- Use computing tools and techniques in other disciplines and understand the effects of this integration for further academic study
- Be prepared to proceed to graduate level work in multimedia and related disciplines or enter a technology-rich work environment
- Be aware of intellectual property issues as they apply to multimedia

Appendix 2: Resources and Advice

Know the Job Situation

Figure out the job situation in your field and be honest about your chances. This doesn’t mean you shouldn’t continue if there are few jobs; it just means you should know what you will be facing. Peter Conn’s *Chronicle of Higher Education* piece “We Need to Acknowledge the Realities of Employment in the Humanities” is a good place to start, as are Bethany Nowviskie’s “The #alt-ac Track” and the Media*Commons* project she edits, *#Alt-Academy*.

Know the Expectations in the Field You Want to Pursue

As mentioned in the chapter, many government, educational and industry groups have developed descriptions of the skills, literacies and competencies they expect or hope for. Read those and ask whether you have the skills and how you demonstrate competency in your CV. Some examples of competencies are:

- New Media Competencies
- http://www.culturalhrc.ca/minisites/New_Media/e/01-02-01.asp
- The New Media Literacies
- http://www.newmedialiteracies.org/the-literacies.php
- Researcher Development Framework
- http://www.vitae.ac.uk/policy-practice/234301/Researcher-Development-Framework.html
- Key Leadership Competencies
- http://www.tbs-sct.gc.ca/tal/kcl/intro-eng.asp

Read the Job Ads Long Before

Look for advertisements for the types of jobs you would like and pay careful attention to the competencies and requirements they list. Do this before the moment that you desperately need a job. Job ads inform you about the field and the expectations of employers. Ask yourself how you can get to the point where your CV would show that you are suitable for the jobs you would like. Here is an example of a job ad posted to *Humanist* in 1997 (*Humanist Discussion Group* 10.752) that helpfully describes the competencies the organization is looking for:

> **Project Manager**
>
> The Getty Information Institute is a leader in promoting innovative and effective uses of information technology in the arts and humanities.
>
> The Institute is looking for a full-time professional with strategic skills to develop and manage projects that promote worldwide access to cultural heritage information. The successful candidate will have creative, administrative and financial responsibility for a range of activities involving digital imaging, interoperability, data standards, intellectual property rights, information policy and training. Typically will lead several simultaneous projects, act as an in-house advisor in the area of digital imaging and conduct or supervise research. Will oversee any special conferences, symposia, workshops, and/or publications related to project activities.
>
> Requirements include a graduate degree in the arts, humanities, or information science, or equivalent; 8–10 years experience, including management of complex projects with technology components; digital imaging expertise; excellent oral and

written communications skills. International experience and a foreign language are highly desirable.

Follow Sites about Professionalization and Project Management

There are a number of curated web sites where you can get advice about becoming a professor or managing projects. The *Chronicle of Higher Education* has an edited and multi-authored blog with lots of advice called *ProfHacker*(http://chronicle.com/blogs/profhacker/). The Association for Computing in the Humanities (ACH) runs a *Digital Humanities Questions & Answers* site where you can post questions or just read the discussion around those of others. They have a section on project management and digital humanities professions (http://digitalhumanities.org/answers/forum/project-management).

Get Advice from Others at the Start

There is good advice out there for new graduate students, but that advice can make the most difference if you pay attention early on. It is much easier to weave activities in that enhance your portfolio if you think about it early and can opportunistically add activities as you go. It is much harder to try to jam all the professionalization in at the last moment. A great place to start if you are a new graduate student in the humanities is Brian Croxall's "An Open Letter to New Graduate Students," *Chronicle of Higher Education*, August 19, 2010, http://chronicle.com/blogs/profhacker/an-open-letter-to-new-graduate-students/26326.

Your professors, fellow students and supervisor are also good sources of advice. Career counseling centers are full of advice, most of it too sensible to stand.

Remember, however, not to trust any one source of advice. If your supervisor tells you not to worry about professionalization and just concentrate on the thesis you need to ask whether that is the right path for you (and it may be.)

Get Experience in Digital Humanities Projects

Seek out digital projects that you can contribute to. Few digital humanists graduated from programs that prepared them in computing in the humanities. Most fell into it or sought out project work that provided them an apprenticeship in the field. If you want to get experience and if you want to be able to demonstrate experience then find and volunteer to work on such projects. If there aren't curricular opportunities then try the following:

- See if the library is digitizing materials and if you can participate in that.

- See if your university has undergraduate or graduate research opportunities where you could propose your own project and be paid to pursue it. Alternatively, you could propose an independent study.
- Introduce yourself to faculty and staff who seem to have projects and see if they need someone with your skills. If they have regular meetings ask if you can sit in to see if there is a fit. To figure out who has projects, check out the website or news feed of your faculty.
- See if you can get a job in the computer store or computing services and learn from the job.

Learn How to Program

One of the skills in greatest demand is programming as it makes digital things work. If you think you would like to program then try to learn one or more languages. Learning to program is a great thing to do in school because you have access to courses, there are lots of people to help you and you have time. Your first language will take you six months to learn to the point where you can build stuff easily. As for which language to learn, choose the one that attracts you and for which you can get friendly help. The Association for Computing Machinery (ACM) has a magazine for students called *XRDS Crossroads* (http://xrds.acm.org) that has advice, including an article by Ben Deverett on “How to Learn Programming Languages” (http://xrds.acm.org/resources/how-to-learn-programming-languages.cfm).

Acquire Other Technical Skills

There are a number of other technical skills that are in demand. The ability to solve computing and networking hardware problems is always in demand. The ability to run server systems is in great demand as more and more of our projects go onto servers. Good graphic design skills combined with an understanding of web technologies are always needed by web projects. Many digital humanities projects need people who can edit XML or work with GIS tools.

Join a Community of Inquiry or Create One

A great way to get involved in the field is to join a community whether it is a mailing list like *Humanist* (http://digitalhumanities.org/humanist/), a local working group or an international association like the Association for Computing in the Humanities (http://ach.org/). Most organizations are looking for keen volunteers and through volunteering you can meet people and informally learn the tacit knowledge of the field.

Some organizations organize events that bring us together in face-to-face meetings like seminars and conferences. Go to the meetings you can, because much of the breaking research is reported at conferences and not in publications. Much of the scholarship in this field is digital, which is shown before being written up. Conference presentations are where you can find out about how digital works were made and ask questions. One annual conference that is graduate student friendly is that of the Canadian Society for Digital Humanities or CSDH-SCHN (http://csdh-schn.org/), which has meetings each year somewhere in Canada as part of the Congress of the Humanities and Social Sciences. If you don't have many local community resources, access to conferences, or a community to learn from, you should try to kick-start your own community by organizing events. One type of event that builds community is an "unconference." An unconference is one where the participants decide the agenda and teach each other. An unconference is not your typically collection of expert talking-heads who lecture you from the pulpit; rather, it is designed to encourage participants to share what they know. An unconference can be organized by students for students. The point is that you don't need anyone authorizing you to start doing stuff. For a tested model of an unconference, see the THATcamp web site (http://thatcamp.org/).

An alternative approach is to join an industry association, especially if you know the industry you want to be part of. For example, if you want to work in technical writing and communication and you are in British Columbia, the Society for Technical Communication–Canada West Coast (http://stcwestcoast.ca/) have volunteer opportunities, job seeking events, meet-ups and guides. Such associations typically have student membership rates and their meetings can be a good way to network with potential employers.

Above all, enjoy what you do.

Footnotes

1 John Guillory, "Professionalism: What Graduate Students Want," *Profession* (1996): 91–99 (91). It should be noted that Guillory is critical of pre-professionalization. We will return to him in our conclusion.

2 Felice J. Levine, "Professionalization, Certification, Labor Force: United States," in *International Encyclopedia of the Social and Behavioral Sciences*, ed. Neil J. Smelser and Paul B. Bates (Oxford: Elsevier, 2001), 12146.

3 We prefer acculturation, as it will become clear that we have in mind something broader than just preparing students for jobs. Acculturation is preparing students so they fit in the culture of a field that may span many different types of jobs. Nonetheless, we find ourselves using the word professionalization interchangeably.

4 There was a flurry of articles in *The Chronicle of Higher Education* in April 2010 that dealt with the lack of jobs for PhDs in the humanities: Peter Conn set out the dismal data and proposed

recommendations including programmes being more open to alternative jobs; Diane Auer Jones suggested preparing students for a broader spectrum of jobs, and Katharine Polack called for ideas. See Peter Conn, "We Need to Acknowledge the Realities of Employment in the Humanities," *The Chronicle of Higher Education*, April 4, 2010, http://chronicle.com/article/We-Need-to-Acknowledge-the/64885/; Diane Auer Jones, "Are the Humanities Dead, or Are Academic Programs Just Too Narrow?" *The Chronicle of Higher Education*, April 9, 2010, http://chronicle.com/blogs/brainstorm/are-the-humanities-dead-or-are-academic-programs-just-too-narrow/22454; and Katharine Polack, "A Letter from a Graduate Student in the Humanities," *The Chronicle of Higher Education*, April 4, 2010, http://chronicle.com/article/A-Letter-From-a-Graduate/64889/.

5 Frank Donoghue goes further and suggests that there is something self-serving in the number of students admitted into graduate programs; that we need them to "provide teachers for their lower-division courses (particularly first-year writing sections) as cheaply as possible" ("An Open Letter from a Director of Graduate Admissions," *The Chronicle of Higher Education*, April 4, 2010, http://chronicle.com/article/An-Open-Letter-From-a-Director/64882/). One could add that this explains what professionalization we do provide—we train them to do the work we don't want to do, like teaching undergraduates.

6 See Conn, "We Need to Acknowledge," and Jones, "Are the Humanities Dead."

7 Conn, "We Need to Acknowledge."

8 If you want a sense of the breadth of digital humanists, you can look at the list of contributors to *#alt-academy: Alternate Academic Careers*, ed., Bethany Nowviskie, MediaCommons, http://mediacommons.futureofthebook.org/alt-ac/. Alternatively, you can look at the contributors to the Day of Digital Humanities over the last three years at http://tapor.ualberta.ca/taporwiki/index.php/Day_in_the_Life_of_the_Digital_Humanities.

9 Bethany Nowviskie defines #alt-ac jobs as "a broad set of hybrid, humanities-oriented professions centered in and around the academy, in which there are rich opportunities to put deep—often doctoral-level—training in scholarly disciplines to use" ("The #alt-ac Track: Negotiating Your 'Alternative Academic' Appointment," ProfHacker, *The Chronicle of Higher Education*, August 31, 2010, http://chronicle.com/blogs/profhacker/the-alt-ac-track-negotiating-your-alternative-academic-appointment-2/26539).

10 For a discussion of the digital humanities as building, see Stephen Ramsay's blog entry "On Building," *Stephen Ramsay*, January 11, 2011, http://lenz.unl.edu/papers/2011/01/11/on-building.html, and the comments to the post.

11 Geoffrey Rockwell presented a short version of this argument for acculturation at a conference on "A Vision for Digital Humanities in Ireland" organized by the Digital Humanities Observatory of the Royal Irish Academy (Dublin, March 31, 2011). For more details, see his report, "Conference Report on the DHO conference A Vision for Digital Humanities in Ireland," *philosophi.ca*, April 20, 2011, http://www.philosophi.ca/pmwiki.php/Main/DHOAVisionOfDigitalHumanitiesInIreland.

12 Agile programming is essentially a methodology that emphasizes short, iterative cycles of development in close collaboration with a client.

13 There are many tools out there and many lists of project management tools. A good place to start is Cameron Chapman's "15 Useful Project Management Tools," *Smashing Magazine*, November 13, 2008, http://www.smashingmagazine.com/2008/11/13/15-useful-project-management-tools/.

14 Stéfan Sinclair helped start the Humanities Computing MA at the University of Alberta and was then recruited to McMaster where he teaches in the Multimedia program. Geoffrey Rockwell developed the Multimedia undergraduate program at McMaster and was later recruited to the University of Alberta where he teaches in the Humanities Computing MA and has been the graduate coordinator. We continue to work well together, especially at this distance.

15 Originally, students could only take Multimedia in a double major combination with another subject; now they can do a single honors. For more on the current program, see http://www.humanities.mcmaster.ca/undergraduate/multimedia.html.

16 For more on the development, rationale and naming of the program, see Geoffrey Rockwell, "Is Humanities Computing an Academic Discipline?" (paper presented at the University of Virginia, Charlottesville, November 1999), http://jefferson.village.virginia.edu/hcs/rockwell.html, and "Multimedia, is it a Discipline? The Liberal and Servile Arts in Humanities Computing," *Jarhbuch für Computerphilologie* 4 (2002): 59–70. We deliberately decided not to call it a Humanities Computing program, as we did not think that would communicate appropriately to prospective undergraduate students.

17 The way this definition plays out is explained in Rockwell, "Is Humanities Computing an Academic Discipline." For more reflections on multimedia and its teaching, see also Geoffrey Rockwell and Andrew Mactavish, "Multimedia," in *A Companion to Digital Humanities*, ed. Susan Schreibman, Ray Siemens, and John Unsworth (Malden: Blackwell, 2004), 108–20, and Andrew Mactavish and Geoffrey Rockwell, "Multimedia Education in the Arts and Humanities," in *Mind Technologies: Humanities Computing and the Canadian Academic Community*, ed. Raymond G. Siemens and David Moorman (Calgary: University of Calgary Press, 2006), 225–43.

18 One constant challenge is to ensure that all students working collaboratively develop a core set of competencies—it can be too easy for students to specialize and continue honing acquired skills, rather than investing the time and effort in developing new ones. We have found that mixed evaluation methods (individual and group work) can be effective for this.

19 This has led this approach to be described as "reconstituted behaviorism" in a critical article by Terry Hyland, "Competency, Knowledge and Education," *Journal of Philosophy of Education* 27, no. 1 (1993): 57–68. He concludes, "Competence-based approaches to education have a weak and confused conceptual base, are founded on dubious and largely discredited behaviorist principles, and display systematic ambiguity in their treatment of knowledge and understanding" (66). The problem with Hyland is that ambiguity and conceptual differences are true of any principles used in curricular design. "Knowledge" and "understanding" are notoriously ambiguous and contested concepts. Ultimately we have to use the concepts at hand with their weaknesses and traditions of use while being willing to critique the very tools we use.

20 See, for example, the "New Media Competencies" of the Cultural Human Resources Council of Canada, http://www.culturalhrc.ca/minisites/New_Media/e/01-02-01.asp, or "The New Media Literacies" of the New Media Literacies project, http://www.newmedialiteracies.org/the-literacies.php.

21 They were used, for example, in a five-year retreat when, with a full faculty complement, we reviewed the curriculum.

22 See the Humanities Computing website, http://humanitiescomputing.ualberta.ca, for information about the program. For a fuller account of the development of the MA in Humanities Computing at Alberta, including additional details on the curriculum, see Stéfan Sinclair and Sean Gouglas, "Theory into Practice: A Case Study of the Humanities Computing Master of Arts Programme at the University of Alberta," *Arts and Humanities in Higher Education* 1, no. 2 (2002): 167–83.

23 Matthew Kirschenbaum explains how the term “digital humanities” originated in 2001 during discussions with Blackwell Publishing about what would become the *Companion to Digital Humanities*. See “What Is Digital Humanities and What’s It Doing in English Departments?,” *ADE Bulletin* 150 (2010): 55–61.

24 For a discussion on this relationship, see Simon Mahony and Elena Pierazzo’s chapter, “Teaching Skills or Teaching Methodology?”

25 On teaching text analysis, see our chapter, “Teaching Computer-Assisted Text Analysis: Approaches to Learning New Methodologies.”

26 For a discussion on teaching programming in the humanities, see Stephen Ramsay, “Programming with Humanists: Reflections on Raising an Army of Hacker-Scholars in the Digital Humanities,” another chapter in this collection.

27 For more details on our efforts to balance technical and theoretical skills, see Sean Gouglas, Stéfan Sinclair, and Aimée Morrison, “Coding Theory: Balancing Technical and Theoretical Requirements in a Graduate-Level Humanities Computing Programme,” in *Mind Technologies: Humanities Computing and the Canadian Academic Community*, ed. Raymond G. Siemens and David Moorman (Calgary: University of Calgary Press, 2006), 245–56.

28 For an introduction to problem-based learning, see James Rhem, “Problem-Based Learning: An Introduction,” *National Teaching & Learning Forum* 8, no. 1 (1998): 1–4.

29 It should be noted that this is the way Rockwell has organized the course. Other instructors, depending on the number of students in the class, will have all the students work on one project for CSL or have them develop their own projects. Regardless of the organization, we believe the learning is similar.

30 Approximately one in ten students already has a job and has negotiated flex-time to complete the MA part-time. These students are typically mature students, already integrated into a profession.

31 Not all of the research assistantships are in digital humanities projects. In some cases they are brought into larger teams working on projects in the humanities that have a digital dimension. Thus, they might be working with other English research assistants on a digital edition or with philosophy research assistants on the web site for an ethics project.

32 A separate issue is how to gracefully correct and encourage quality in graduate student work. This sort of apprenticeship is a very different relationship than marking an assignment that has no real purpose beyond assessment. We believe it is important to the spirit of collaborative work that problems in the work be addressed immediately, and the student then asked what help they need to meet a standard of quality suitable for publishable research. The key is that you need to get to the point where you trust their work enough to use it—then you are collaborators, and that is an important transition.

33 For an introduction to this method, see Jared M. Spool, “The Essence of a Successful Persona Project,” *User Interface Engineering*, February 17, 2010, http://www.uie.com/articles/essence_personas/, and then look at Alan Cooper, *The Inmates Are Running the Asylum*(Indianapolis: Sams Publishing, 2004).

34 See Paul Andrew, “10 Completely Free Wireframe and Mockup Applications,” *Speckyboy Design Magazine*, January 11, 2010, http://speckyboy.com/2010/01/11/10-completely-free-wireframe-and-mockup-applications/.

35 Another advantage of this process is that you secure consensus among stakeholders at each step, so you are not forced to redevelop the interface because someone objects to an interface at the last moment. It can be hard to negotiate interface when stakeholders are presented with finished designs and have no stake in the process that generated the final design.

36 A starting point is the Graduate Studies office of your university. They have to deal with dysfunctional supervision in all its forms and will typically have resources, workshops, and advice for both supervisors and students. If your university does not have anything of this sort, we suggest consulting Heather Latimer's *Literature Review on Graduate Student Supervision* (prepared for the Dean of Graduate Studies Task Force on Graduate Student Supervision at Simon Fraser University, 2005), http://www.sfu.ca/uploads/page/06/lit_review_.pdf.

37 Guillory, "Professionalism," 97.

38 Jennifer Wicke, "I Profess: Another View of Professionalism," *Profession* (2001): 52.

39 The report is freely available on the MLA website, http://www.mla.org/professionalization.

40 Guillory, "Professionalism," 98.