The Sociological Quarterly ISSN 0038-0253

INTERPROFESSIONAL RELATIONS AND THE EMERGENCE OF A NEW PROFESSION: Software Engineering in the United States, United Kingdom, and Canada

Tracey L. Adams*
University of Western Ontario

This article explores the conditions under which interprofessional relations may be characterized by conflict and/or cooperation through a case study of the rise of a software engineering (SWE) profession in the United States, United Kingdom, and Canada. Analysis of professional journal publications produced by the primary computing organizations in all three countries reveals that the development of a SWE profession in the United Kingdom, and to a lesser extent, the United States, has been a collaborative project pursued by engineering and computing organizations. In contrast, the development of SWE in Canada has been characterized by intense interprofessional conflict. The investigation suggests that interprofessional conflict may be mitigated when professionalizing groups seeking to lay claim to the same jurisdiction are similar in terms of their organizational size, outlook, and regulatory status.

One of the fastest growing occupations in the Western world is that of "software engineer" (Bureau of Labor Statistics 2001). Although the term "software engineering (SWE)" was coined over 30 years ago, it was not used regularly to refer to a distinct set of computing-related tasks until much more recently. Despite its youth, SWE is a rapidly professionalizing occupation. In the past 20 years, occupational leaders from around the globe have endeavored to define SWE as professional work by formalizing education, delineating a knowledge base and scope of practice, establishing codes of ethics, and seeking to raise its status. The registration and licensing of software engineers has become more common. While the professionalization of SWE is occurring on an international level, the dynamics of its development have differed significantly from nation to nation. Notably, in Britain, engineering and computing organizations have worked together to establish education and registration criteria for SWE, and to further its status as a computing and engineering profession. The development of SWE in the United States has also been a project largely shared between computing and engineering organizations, although some conflict and tension has been evident. In Canada, however, SWE has become the subject of intense conflict between engineering and computing groups. Thus, in some contexts, the development of SWE appears to have been fostered by interprofessional cooperation, while in others it is an arena for interprofessional conflict.

This article explores the divergent development of SWE in the United Kingdom, the United States, and Canada to document the emergence of a new profession and to

*Direct all correspondence to Tracey L. Adams, Department of Sociology, University of Western Ontario, London, ON N6A 5C2, Canada; e-mail: tladams@uwo.ca

illuminate the role of interprofessional conflict and cooperation in processes of professionalization. The theoretical and empirical literature on interprofessional relations has focused on conflict, to the neglect of cooperation. Nonetheless, the case of SWE demonstrates that, in certain contexts, cooperation among separate occupational groups is not only possible, but crucial to a professional project. This study strives to enhance our theoretical understanding of interprofessional relations by identifying conditions that encourage interprofessional cooperation. Specifically, it contends that when neighboring occupations are well organized and have a similar legislative status[1] and outlook, cooperation is possible. When differences in organization, outlook, and/or status are present, conflict is more likely to occur.

INTERPROFESSIONAL RELATIONS AND PROFESSIONAL DEVELOPMENT

SWE is one of many occupations undergoing professionalization. Although professionalization has been variously defined, it is most basically understood as a set of processes through which occupational leaders seek greater status and privileges for occupational practitioners. These processes and privileges can differ across time and place, but in the Anglo-American model of professionalization—characteristic of the United States, United Kingdom, and Canada—occupational leaders appeal to the state for work privileges (such as a claim to a monopoly in the performance of certain tasks and the privilege to use a certain work title) and to the public for social esteem and patronage (Larson 1977; Collins 1990; Freidson 1994, 2001). Claims to professional status and privileges are based on claims to expertise: that is, the work in question is said to require a great deal of skill and knowledge if it is to be performed in a manner that does not endanger the public (Larson 1977; Freidson 2001). In general, the most successful professional projects have been those pursued in the 19th and early 20th centuries by groups such as medical doctors and lawyers. Other groups like dentists, veterinarians, and optometrists have also been quite successful in winning legislation granting them privileges, including a protected title (no one who is not approved by a professional body can claim to be a dentist, veterinarian, etc.), self-regulation (the ability to govern themselves, albeit with an increasing amount of outside involvement), and especially for dentists and veterinary doctors, a virtual monopoly in a given practice area.

However, the granting of such extensive professional privileges is becoming rare. While in the past, governments appear to have been willing to grant fairly extensive professional privileges to a relatively small number of professional groups, increasingly they are granting much more limited powers to a wider number of claimants (Coburn 1999). As more occupational groups seek professional status and with the changing nature of professional employment, the benefits of professionalism appear to be decreasing (Coburn 1999; Leicht and Fennell 2001). Nonetheless, many occupational leaders continue to seek greater professional status in the hopes of increasing their job security, autonomy, social esteem, and income. Their efforts to do so, however, are hampered, like those of their predecessors, by the actions of more established professions seeking to maintain an existing status and other aspiring professions' projects to secure their own places in the labor market and society more broadly (Willis 1983; Freidson 1994; Adams 2004a; Kelner et al. 2004). Interprofessional conflict is the outcome.

For Abbott (1988), it is this interprofessional conflict that drives professional development and change. Abbott is critical of the traditional literature on professionalization, arguing that it has focused too much on the patterns or stages of professionalization, and hence, yielded little useful information. Rather, Abbott (1988:19–20) recommends a shift in focus away from studying how single occupations acquire the trappings of professional status, toward examining the content of professionals' work, and professions' efforts to establish and maintain a *jurisdiction*, which Abbott defines as the link between a profession and its work. Central to Abbott's argument is the idea that professions exist in a broader system wherein their jurisdictions or scopes of practice meet and sometimes overlap. Individual professions seek to establish their own jurisdiction—their claim to perform certain tasks and exercise authority in a given area—often in the face of competition from other groups of workers eager to claim the same or similar tasks for themselves. Each occupation attempts to hold a jurisdiction by claiming a prestigious body of knowledge and appealing to third parties—the public, the state, and the workplace (pp. 53–54, 59–69). Interprofessional conflict occurs when two or more occupations attempt to claim the same jurisdiction. Abbott sees such conflict as endemic to the system of professions.[2]

Conflict between professions is generally resolved in one of two ways, according to Abbott (1988:69): (1) one occupation becomes dominant, subordinating others; and (2) the jurisdiction is split, by task or by client, and divided among the competing occupations, or is sometimes shared without being strictly divided. When two or more professions have overlapping jurisdictions, there are, occasionally, other outcomes. For instance, Abbott (1988:95) mentions "enclosure" where members of different professions who do similar work break away to form their own independent professional group. In a similar vein, separate groups may "amalgamate" to form one unified group. Abbott says amalgamation rarely works; giving the example of the American engineering profession, he argues that "distinct professional heritages and tasks prevent a unified cognitive and social structure" (p. 105).

Abbott's theoretical model of the system of professions and interprofessional conflict has been quite influential, prompting a plethora of studies exploring how interprofessional conflict over jurisdiction has shaped the development of professions historically (Willis 1983; Witz 1992; Coburn 1994), and in recent years (Hafferty and Light 1995; Norris 2001; Hartley 2002; Leveille 2002; Timmermans 2002; Kelner et al. 2004). These studies have underscored the importance of interprofessional conflict to professions and to professionals' experiences of their work. Some have moved beyond Abbott to explore the significance of interprofessional conflict to professionalization (Coburn 1994; Adams 2004a; Kelner et al. 2004).

Despite the considerable evidence supporting Abbott's arguments, especially in the literature on health care professions, his model is limited by its failure to conceptualize a full range of interprofessional relationships. Abbott (1988:135) took "interprofessional conflict" as the phenomenon to be explained and, as a result, failed to theorize moments of interprofessional cooperation, with the exception of his brief discussion of enclosure and aggregation.[3] Furthermore, because he assumed that interprofessional conflict would occur, he did not endeavor to identify the contextual factors that might either mitigate conflict or exacerbate it.

This article seeks to refine Abbott's model by considering those factors that shape relations between neighboring occupations seeking to lay claim to the same jurisdiction. The occupationally crowded information technology (IT) field makes an ideal case study for this endeavor. Abbott (1988) himself wrote on this field, arguing that there was no single occupation capable of jurisdiction, despite many potential claimants. In a more recent study, Ensmenger (2001) has discussed professional activity among various groups of computing workers, documenting both cooperative efforts to professionalize the field, and the differing visions of professionalism that continue to divide it. It is in this context of both conflict and cooperation that a SWE profession is emerging. Nevertheless, an international comparison of professional developments in SWE reveals that conflict has been more prevalent in some locales than others. By comparing the trajectories of professional development in SWE in the United States, the United Kingdom, and Canada, we can identify certain conditions that appear to encourage interprofessional cooperation and mitigate interprofessional conflict.

In identifying the conditions shaping interprofessional relations, it is helpful to return to Abbott and the sociological literature on professionalization more generally. Abbott (1988:82–83) argues that success in claiming jurisdiction is influenced by a group's level of organization, and its ability to obtain cultural and social legitimacy (pp. 60–64). These factors have also been identified as being important to successful profession creation (Larson 1977; Starr 1982). Also relevant are processes of social closure whereby occupational leaders restrict access to education, skills, and entry to practice (Murphy 1988). Organization, cultural legitimacy, and social closure all contribute to professional power, and hence, may shape occupational leaders' ability to claim jurisdiction, even in the face of opposition. Nevertheless, as Abbott (1988:140) contends, many jurisdictional disputes involve professions similar in size and power. In such cases, I suggest, cooperation may be more likely to occur.

The following analysis of professional development and interprofessional relations in SWE suggests that when two occupational groups have similar levels of power—in terms of their social closure, organization, and cultural and legislative status—and a similar outlook, they are more likely to cooperate as each will be unable to enforce its will on the other and claim a jurisdiction for itself. When occupational groups are less evenly matched in terms of power, status, and organization, jurisdictional conflict is more likely to occur.

SOURCES AND METHODS

This analysis of the emergence of SWE in Canada, the United Kingdom, and the United States sprang from a broader project exploring professional developments in computing in these nations since the 1950s. Document analysis of the principal professional journals published by the most prominent computing-related professional organizations in the three nations of focus was conducted. All articles touching on areas of professional development and change, broadly defined (and including issues in computing as a job, a profession, a scholarly endeavor, etc.) were examined.[4] Technical discussions of computing practice were generally omitted, although when debates over techniques turned into discussions of professional roles, they were considered. In all, about 375 articles and an additional 210 letters published in journals across all three nations from 1980 to 2000 were included in this analysis.[5] Additional reports published on the Internet by the key associations studied here were also examined.

The principal sources of information on computing in the United Kingdom were two journals published by the British Computer Society (BCS): their *Computer Bulletin* (1980–2000) and the *Software Engineering Journal* (published jointly with the Institute of Electrical Engineers [IEE] between 1986 and 1996).[6] Additional Web publications by the BCS and engineering organizations like the IEE were also considered. For information on professional activities in the United States, the primary source was the *Communications of the ACM*, a journal published by the American-based Association for Computing Machinery (ACM) since the late 1950s, which claims to be the "internationally acknowledged premier magazine of the computing field" (ACM 2005). Additional ACM publications relating to SWE published on the Internet also informed the analysis. The main professional IT organization in Canada is the Canadian Information Processing Society (CIPS). CIPS has a somewhat checkered publication history, producing four different journals successively between the 1960s and the early 1990s. The result is a gap in the computing literature on SWE in Canada in the mid-1990s. Nevertheless, CIPS's stance on the emergence of SWE has been carefully and formally articulated in its Web publications. Additional information on SWE in Canada was found in the Web publications of the Canadian Council of Professional Engineers (CCPE), the Professional Engineers of Ontario (PEO), and the Canadian Association of Computer Science (CACS).[7]

Together, these document sources provide a wealth of information on professional developments in the IT and computing fields. Each of the journals was a key medium through which national associations communicated with their membership and the public about professional issues and trends. While the publications have tended to focus primarily on computing work and workers, they also provide information on the emergence of SWE, and, to a lesser extent, relations with engineering organizations. A document analysis of these key publications reveals professional developments in computing leaders' approaches and attitudes toward professionalism, and the interprofessional relations shaping the emergence of a SWE profession.

This present analysis was driven by three central research questions. First, what were the factors and trends contributing to the emergence of SWE as a distinct occupation in the United Kingdom, the United States, and Canada? Second, given that SWE is linked with both engineering and computing work, what role have interorganizational and interprofessional relations played in shaping professional development internationally? Third, how can we account for the international differences in interprofessional relations and their impact upon professional development, in the three nations under investigation—nations, which have long been typified as similar in terms of professional development (Collins 1990; Evetts and Buchner-Jeziorska 2001)?

In the discussion that follows, I outline how the development of a SWE profession differed in these three nations, despite their similarities, and I explore the role of organization and regulatory status in shaping interprofessional relations.

PROFESSIONAL ACTIVITIES IN COMPUTING AND THE SOFTWARE CRISIS

Before we can turn to an examination of the emergence of SWE as a distinct occupation, it is necessary to consider the context in which it developed. While a number of factors have shaped the rise of SWE, I will discuss two that are particularly important: (1) earlier professional developments in computing in the United Kingdom, Canada, and the United States; and (2) the expansion of the software industry and growing concerns about software quality.

Early Professional Developments in Computing

There has been professional activity in the field of computing since the birth of modern electronic computers in the years following World War II. The first professional organizations in computing arose in the United States as those university scientists and engineers active in constructing and programming the first computers wanted a forum to interact and share ideas with each other. The ACM was established in 1947, and computing groups within American engineering organizations were established around the same time.[8] By the late 1960s, these organizations had expanded both their membership and their mandate—becoming more interested in advancing computing as a profession and a discipline (Orden 1967:145; Ensmenger 2001). While the ACM is an American-based organization, it has an international membership including a strong presence in Canada and in the United Kingdom. Professional organizations in computing emerged later in the United Kingdom and Canada, perhaps reflecting the slower development of a computing industry in these nations. The BCS was established in the late 1950s in a conscious effort to bring together computing academics, scientists, and business users of computers. In the late 1960s, BCS decided to become a “professional” group, and in the ensuing years, worked to establish credentials and membership criteria that would define BCS members as professionals. Like its British counterpart, the Computer and Data Processing Society of Canada was established in the late 1950s, uniting scientists and business users. In the 1960s, this organization changed its name to CIPS, and debated pursuing a professional mandate (through raising membership standards and establishing credentials); it made little progress at first, however, as a slight majority of members were opposed.

Each of these three organizations took somewhat different paths. BCS was the most explicitly “professional” association, committed to enhancing the professional standing of its members and raising the status of its profession. Membership in the BCS was based on education, experience, and examinations. Those who had met these tests of competence could claim the credential "MBCS" (member of the BCS), and in this way distinguish themselves from the mass of workers in computing-related jobs. In 1984, BCS was granted a royal charter conferring on it a limited number of professional privileges. In contrast, ACM was less committed to pursuing professionalism through establishing credentials and seeking legislation. Rather, this organization endeavored to define computer science as a legitimate scientific discipline and university subject. Other American-based organizations, such as the Data Processing Management Association, placed greater emphasis on establishing credentials and examinations for data processing and computer workers (Ensmenger 2001). The Canadian organization, CIPS, has been less committed to a professional project than organizations in the other two nations (Adams 2004b). Until the 1980s, it remained a cluster of local societies (bringing together practitioners to discuss practical problems) linked by a national executive. During the 1980s, CIPS's leadership formally sought greater professional status through activities such as accrediting education programs, establishing a voluntary credential available to members, and pursuing recognition through government legislation.[9]

While these organizations followed different paths, they tended to face the same difficulties and obstacles that both encouraged the desire of many computing workers to professionalize, and hampered their ability to do so. Primary among these was the makeup of the computing and data processing workforce. From the first, computing-related employment was not concentrated in a few select jobs, but rather entailed a wide range of occupations. With the rapid expansion of computing-related employment in the 1960s, 1970s, and 1980s, the computing workforce was increasingly made up of a heterogeneous group of people who shared neither an educational background nor an occupational focus. Hence, there was no single computing worker or occupation to professionalize. Professional projects in computing have been somewhat unusual in that organizations have attempted to raise the status of computing workers generally, rather than a single occupation with a defined scope of practice and knowledge base (Adams 2004b).

Discussions of the "computing profession" in the early publications of the ACM and BCS were inherently vague. When a specific occupation was mentioned, it was most often the "professional programmer" who was singled out, particularly in the United Kingdom (Orden 1967; Barron 1975). However, no specific scope of practice or knowledge base for these computing professionals was evident, although members of the BCS and ACM did struggle to delineate one. Despite their efforts, the computing body of knowledge remained vague enough that in 1969, CIPS (1969:8) described it only by asserting that "the computer is at the heart of it" as was the "method of using it [the computer], i.e. programming." Moreover, while computing professionals remained a rather amorphous group with a largely undefined set of skills, computing leaders were eager for professional advancement because of the rapid expansion and changes in the field. Skilled computer-related workers continually sought ways to demonstrate their expertise (which often was informally acquired), and to distinguish themselves from the mass of lesser-trained workers entering the field. This commitment to professionalism was renewed in the 1980s in the context of rising social concern over the effectiveness of software practice and production.

The Software Industry and the "Software Crisis"

The software industry was extremely small until the late 1960s (Johnson 1998). In the 1950s and 1960s, dominant computer companies like IBM bundled software with the hardware they sold. Accompanying both the software and the hardware was a strong customer service commitment. There was little market demand for additional software, and IBM was willing to develop software upon customer request (Johnson 1998). Industry growth in software was largely spurred by IBM's decision in 1969 to unbundle software from hardware. The industry expanded rapidly in the succeeding decades, fueled by the growing application of computers to more areas of social life.

Concerns about software quality expanded along with the industry. These concerns were evident in the 1970s (Hoare 1975), but increased in the 1980s, until many spoke of a "software crisis" (Shore 1988; Chapman 1990). By and large, this was a crisis of quality. Journal articles complained that software failed too often, and was often more costly to produce than predicted (Hoare 1975; Shore 1988; Chapman 1990; Denning 1990, 1998). Customer satisfaction was said to be low. Concerns within the industry were partly driven by vocal complaints outside of it. Computer failures and mistakes that affected service provision in a variety of sectors were invariably blamed on poor software (Kocher 1989; Chapman 1990; Denning 1990). Hacking and computer sabotage were similarly seen to be made possible through software inadequacies, such as "trapdoors" and poor security (McIlroy 1990). The U.S. government published a report in the late 1980s blaming software inadequacies for the failure of some high-profile, high-budget defense technologies (Chapman 1990). Media reports of instances where faulty software design contributed to the malfunctioning of health care technology, which, in turn, led to patient injury and death, heightened concern over the social consequences of faulty software (Collins et al. 1994). To a public already ambivalent about the spread of computers, high-profile cases of security breaches, computer fraud, software failure, and computer viruses added fuel to the fire (Kocher 1989; Chapman 1990).

While many computing leaders felt that public criticism was quite often unfair and unjust, they themselves held that problems clearly did exist. During the 1980s and 1990s, they advocated many possible solutions in the pages of their journals. Many felt that education was not adequately preparing people for the work environment. Various contributors called for a more practical education (more hands-on involvement in the design, production, and maintenance of software in all of its phases) (CIPS 1973; Conway and Hooper 1988; Frailey 1988). Some argued that the computer science curriculum needed to be less practical and more theoretical; in particular, a stronger emphasis on math in the program was advocated (Ralston 1984; Berztiss 1987; McCracken 1987; Gries 1991). More significantly, there were increasing calls for the greater application of engineering methods to computing in general, and software practice in particular. This argument was first advanced in the late 1960s and the early 1970s by a few. For instance, in the *BCS Computer Bulletin*, Hoare (1975:6) argued that software developers and programmers had a lot to learn from engineers who based their designs on "sound mathematical theories and computational techniques," and generated products at a predicted cost and on time, with "few design failures." These sentiments became more common in the 1980s and gained momentum in the 1990s. The solution to the software crisis was increasingly said to be "engineering," and the emphasis on SWE grew accordingly. However, as we will see in the following sections, the rise of SWE as a field and occupation occurred somewhat differently in Canada, the United Kingdom, and the United States.

SWE IN THE UNITED STATES

During the 1980s, there was little mention of SWE in the pages of the *Communications of the ACM*. Indeed, before 1989, only four pieces were published with the terms "software engineering" in their title: one editorial, one article, and two letters to the editor. In the next decade, fully 27 articles were published explicitly on SWE, while scores of others made reference to it. In the early 1990s, SWE as both a set of tasks and a distinct occupation became a key focus within the ACM. During this era, "software engineering" came to be seen as a principal occupation within the computing field, even though the term was ambiguously applied, at first.

In the first decade of its usage, "software engineering" was rarely defined. Quite often, the term was used almost interchangeably with "programming" (for instance, Wagner 1990; Wilkes 1991). Some saw its usage as a rather pathetic attempt to increase this occupation's status (Rickert 1989). While this is a very narrow use of the term, it is a significant one given that, in decades past, some asserted that "programming" was *the* central professional computing occupation. SWE began to replace programming as the computing "profession" most mentioned in the literature. By the 1990s, programming's status as "professional" was being called into question, but the term "software engineering" clearly connoted a set of tasks requiring skill, decision making, and expertise.[10] The SWE term was also used to refer to a broad but ill-defined set of computing tasks; that is, "as an umbrella term of . . . breadth (and, at present, vagueness)" (McCracken 1987:4). At times, the term "software engineering" appeared to refer to the bulk of computing practice, as when a President of ACM, Bryan Kocher (1990), suggested that ACM could benefit from having a more professional sounding name, like the "Society for Software Engineers." The ambiguity in the use of the term, combined with its growing importance within the computing field and ongoing problems in the software industry, spurred professional activity to establish better standards.

To define and develop professional standards, the ACM joined with an engineering organization, the IEEE-CS, its sometimes collaborator in a wide range of computing-related professional projects.[11] In 1993, the two formed a "Joint Steering Committee for the Establishment of Software Engineering as a Profession." This committee's mandate covered three central initiatives: (1) establish a code of ethics for software engineers; (2) establish curricula and accreditation standards for SWE educational programs; and (3) define a core software engineering body of knowledge (SWEBOK). In 1998, the steering committee was superseded by a joint "Software Engineering Co-ordinating Committee" (SWECC) to "foster the evolution of software engineering as a professional computing discipline" (quoted in Notkin, Gorlick, and Shaw 2000:2). Through these committees, engineers and computing professionals active in both the IEEE-CS and the ACM worked together to structure SWE as a profession that lay at the crossroads of their two disciplines. They produced a code of ethics for software practice, published in 1997 (Gotterbarn, Miller, and Rogerson 1997). The joint committees also worked out arrangements for the accreditation of SWE university programs. Initially, SWE programs were accredited slightly differently depending on their location. Those housed in computing departments were accredited by the Computer Science Accreditation Board (CSAB), in cooperation with the engineering board. Those in engineering departments were accredited by the Accreditation Board for Engineering and Technology (ABET) with cooperation from computer scientists. This division of labor was streamlined with the integration of the CSAB into ABET in 2000 (Rada 1999). Currently, the CSAB is a "participating body of ABET" and a leader in "the accreditation of programs in computer science, information systems and software engineering" (CSAB 2004).

Most striking of the joint professional initiatives has been the mammoth, international project aimed at identifying a SWEBOK.[12] Leaders in this endeavor believe that reaching a consensus on a core body of knowledge is crucial to the development of SWE as a profession (Abran et al. 2001). The SWEBOK project aims to identify and outline the body of knowledge for SWE practice; it seeks to characterize "the bounds of the software engineering discipline" and demarcate it from computer science and other disciplines, as well as to "provide a topical access to the Body of Knowledge supporting that discipline" (Abran et al. 2004:xix). It has produced three key documents: a "straw man" version (1998), a "stone man" version (2001), and most recently a final version (2004). According to these documents, SWE is "the application [and investigation] of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is the application of engineering to software" (Bourque et al. 1998:8). SWE, then, is a method or approach that can be applied to a variety of tasks in the software field. The SWEBOK also carefully demarcates SWE from computer science, arguing that the latter is the founding discipline of SWE but remains separate from it. Computer science is to SWE, what chemistry is to chemical engineering: it provides a core basis of scientific principles, which engineers can utilize to construct useful and efficient products (Bourque et al. 1998:11). More specifically, it is argued that "the fundamental goals of computer science and software engineering differ" (Bourque et al. 1998:11). Computer science seeks to extend knowledge in computing, while SWE aims to apply "this knowledge to building software" (Bourque et al. 1998:12). Thus, while SWE stems from computer science and is linked to it, it is rightfully an engineering discipline and profession (Bourque et al. 1998; Abran et al. 2001).

In the United States, computing and engineering leaders have worked together to define SWE as a profession through delineating its scope of practice, and by formalizing education and a knowledge base, accreditation, and a code of ethics. The emergence of SWE in this context appears to have been shaped by interprofessional cooperation. The collaboration between the ACM and IEEE-CS may stem in part from their overlapping memberships (both have engineer and nonengineer members) and their history of cooperation. The two organizations have worked together quite closely on a variety of professional and organizational projects over the years; in the 1980s, there was even some discussion of merging them (Bechtolsheim 1985; Grosch 1987). The fact that the organizations are similar in size (ACM has 80,000 members while IEEE-CS has 100,000) may also have played a role. Cooperation may have been further facilitated by a shared outlook, as some ACM leaders have contended that computing is inherently an engineering discipline (Denning et al. 1989). Despite this history of collaboration, recent events have strained the ties between ACM and IEEE-CS, and there are signs of emerging conflict.

Interprofessional conflict has arisen over the licensing and regulation of software engineers. In the United States, engineers, per se, are covered by licensing legislation in every state. While these laws vary, they typically require licensed engineers to complete a four-year degree from an accredited program, and then pass an eight-hour exam, possess four years of on-the-job experience, and then pass a second exam. Although most engineers are not licensed, licensing tends to be mandatory for those offering services to the public, particularly civil engineers, and for those engaged in “the design of facilities, roads, transportation and construction where design documents must be submitted to state agencies for approval” (Knight and Leveson 2002:88). In contrast, computing and IT workers are not licensed at all. For many engineers, licensing is a reality, and the licensing of software engineers was a logical progression in the development of the occupation as a legitimate engineering discipline. State licensing boards began to consider licensing software engineers in the 1990s and requested help from SWECC and other bodies in drafting standards and criteria. These activities created great controversy among computing workers. Some felt that as licensing was becoming a reality, computing workers had to get involved in order to ensure that standards were not drafted entirely by engineers (Bagert 1999; Frailey 1999). Others, however, continued to protest that licensing was unjust and harmful (see for instance DeMarco 1999).

The debate and concern over licensing software engineers prompted ACM to pull out of SWECC, the joint committee aimed at creating an SWE profession. ACM leaders came to believe that the SWEBOK program was doing little to advance the interests of computing or the public. They feared that “the primary purpose of SWECC and its SWEBOK project was to support the licensing efforts of software engineers” as professional engineers (ACM 2000:3). ACM investigators came to the conclusion that licensing was not feasible or attractive. Although many reservations were highlighted, three in particular stood out (ACM 2001; Knight and Leveson 2002). First, state licensing of engineers tends to be voluntary and uncommon for engineers employed by others—as most software engineers are; hence, state licensing would likely not ensure that people producing safety-critical software were licensed and capable. Safety-critical software is that which has repercussions for the security and well-being of others, and it is ACM's primary concern with respect to software quality. Second, professional engineering licenses do not distinguish between specialties. As a result, there is little, except professional responsibility, to prevent a licensed engineer with little software training from working as a software engineer. Third, for licensing, states require specific education: most notably, engineers must graduate from an engineering-accredited institution. However, currently, most software engineers have computer science backgrounds, and hence, could be ineligible.[13] Moreover, most states require examinations for engineers that test knowledge in engineering subjects that few computer-science-trained people have had exposure too. State licensing then could limit the ability of computing professionals to practice SWE and could disproportionately benefit engineers.

ACM's withdrawal from SWECC and SWEBOK signals growing conflict between engineers and computing professionals active in creating a SWE profession. Nevertheless, upon withdrawing from SWECC, ACM did reaffirm its commitment to "continue to work closely with the IEEE Computer Society on projects that further the evolution of software engineering as a *professional computing discipline*" (ACM 2000:2, italics added). The latter phrasing may be significant given the IEEE-CS's commitment toward making SWE a "legitimate engineering discipline" (Bourque et al. 1998:2; Abran et al. 2001:1).

This emerging conflict appears to stem from the different regulatory status of computing and engineering. Rules governing education and entry to practice in engineering are ensconced in law, and therefore harder to change than the practices common in computing. Hence, "cooperation" for computing organizations may mean adapting to engineering practices (for instance adapting their accreditation policies to bring them in line with engineers'). Reluctant to do this extensively, computing organizations may increasingly distance themselves from cooperative enterprises. Along with its withdrawal from SWECC, ACM has broadened the focus of its professional project. While at times in the 1990s, SWE appeared to be the main occupation in the discipline, ACM has recently restated its "deep commitment to the professionalization of all parts of the IT field, not just software engineering" (ACM 2000). Toward this end, it has launched the Information Technology Profession Initiative to encourage the development of an IT profession. Furthermore, the idea that engineering is the solution to the software crisis has recently been called into question. For instance, de Champeaux (2002:102) maintains that the "concept of software *engineering* has not been helpful, and plausibly has deeply exacerbated the software crisis" (italics in original). Thus, while during the 1990s SWE appeared to be an important occupation around which professional activity in computing could be based, more recently, computing leaders are once again trying to professionalize a cluster of computing and IT occupations; nevertheless, they have not abandoned their cooperative efforts to establish a SWE profession.

The U.S. case suggests that while similarities in organizational size and outlook may contribute to an environment in which cooperation can occur, differences in regulatory status and legal claims to jurisdiction may be a source of conflict when two groups seek to lay claim to the same jurisdiction.

SWE IN THE UNITED KINGDOM

Efforts to define and construct the field of SWE earnestly began earlier in the United Kingdom than they did in the United States. During the 1980s, SWE was the subject of numerous articles in the BCS *Computer Bulletin*, and the subject of a separate BCS/IEE journal. A look at the BCS journals published in the 1980s and 1990s reveals a computing organization eager to define SWE and to work with engineers to advance and professionalize the field. While the success of interprofessional cooperation in the United States has been somewhat limited, cooperation between British engineers and computing professionals has been more successful, and the links between the two occupations have become even closer since the 1980s as the BCS has become an "engineering institution."

As in the U.S. literature, BCS publications in the 1980s were somewhat ambiguous about precisely what SWE was. At times, the term appeared to refer to programming. For instance, Thomas (1987:3) held that during the 1970s and 1980s, the term was "misused, in an attempt to add a false veneer of professionalism to the worthy art of programming." Writers commented that "software engineering" was misleading in this context because programming did not have an engineering basis (Hoare 1981; Thomas 1987). Nonetheless, many argued that programming and software production more generally should "be based on the types of theoretical foundations and practical disciplines that are traditional in the established branches of engineering" (Naur and Randell 1968 in Thomas 1987:174; Hoare 1981). As in the United States, the term was increasingly applied more broadly to the extent that some argued "most degree courses in computing are, or should be, courses in software engineering" since the majority of computer science students will come to work "initially at least, on the production or maintenance of software" (Bott 1989:174). SWE was said to cover "the whole process of software development, starting when the requirements for a product are first identified and ending, many years later when the software is finally allowed to expire" (Brown 1985).

More technical definitions emerged in the 1990s. For instance, Andriole and Freeman (1993:166) characterized SWE as "the systematic application of methods, tools, and knowledge to achieve stated technical, economic, and human objectives for a software-intensive system." Such definitions encompassed a large part of the discipline of computing. Computer science programs altered accordingly. Some U.K. programs decided "to make software engineering the core philosophy around which the undergraduate [computer science] degree would be developed" (Winder et al. 1987:133). With the expansion in the definition of SWE, some felt the need to distinguish more clearly between software engineers and programmers: Lehman (1991:245) asserted that "the difference between the role of programmers and software engineers is fundamental" despite the fact that "sometimes, particularly in small organizations, responsibility for both areas may be vested in single individuals." During the late 1980s and the 1990s, SWE became a prominent area of professional activity in the field of computing.

The growth of SWE and professional activity in this area were spurred by concern that Britain was not as competitive in the computer and software industry as it should be. In the mid-1980s, the Office Advisory Council for Applied Research and Development (ARCARD) established a working group to investigate the status of the U.K. software industry. The group's report, published in June 1986, raised alarm over the state of the industry, finding that it was "small and fragmented," and held a very small share of the world software market (Crinean and Baguley 1986:37; Newton 1986). Reportedly, "the major U.K. software companies [were] much smaller than their major overseas competitors," and must then have a good strategy to compete internationally (Newton 1986:24). Ultimately, the report concluded that "If the U.K. industry is to maintain and improve its competitive position it must also invest heavily in software engineering" (Crinean and Baguley 1986:35). One of the working group's many recommendations was for the Institution of Electrical Engineers (IEE) to expand its involvement in the field of SWE and explore the certification of software engineers. BCS had already been working toward professional certification in SWE and was eager to be part of any such initiative (Newton 1986:25). British computer workers felt that software quality was too low and that the solution lay with a greater application of engineering principles to software practice (Conway et al. 1989:248; Hoffnagle 1991).

The publication of the ARCARD's working report on the software industry gave further impetus to the joint endeavors of the BCS and IEE already underway. The two organizations had organized joint conferences many times since 1959 and had merged their library holdings in 1977. They came to work more closely together during the 1980s as they both sought to professionalize the field of SWE. While SWE largely grew out of the discipline of computing, the IEE had decided in 1982 to enlarge its scope by admitting software engineers as members. According to one contemporary, this decision was motivated by the members' "professional involvement with software, and . . . the recognition of the need to raise professional standards in software engineering" (Pyle 1986:66). The IEE then worked to specify criteria for membership and professional standards—no easy feat in a field so ambiguously defined. In this endeavor, it worked with members of the BCS. Many leaders in both organizations felt that "the IEE and the BCS (being the two institutions currently most concerned with true software engineering) should make a start together" (Pyle 1986:68). Although the two organizations had different accreditation criteria and "somewhat different interests" (Pyle 1986:68), they were able to find enough common ground to work together to formulate SWE standards (Ould and Thewlis 1987), and to define engineering-based computing and SWE curricula (BCS/IEE 1989; Finkelstein et al. 1993), as well as to cohost SWE conferences and copublish the SWE journal. SWE was developed as a computing *and* engineering profession.

Why were electrical engineers and computing professionals in the United Kingdom able and willing to work together to professionalize the field of SWE? Cooperation between the two was likely facilitated by certain similarities between them and the professional groups they represented, as well as the BCS's efforts to establish itself as an "engineering institution." In the United Kingdom, the term "engineer" is one applied to "anyone whose work relates to engineering—particularly manufacture or maintenance" (Engineering Council [EC] 2003). The broad usage of this term has limited professional advance, as "statutory recognition of professions in the U.K. requires the particular functions being controlled by statute to be defined . . . [and] it has not proved possible to isolate the functions that engineers undertake" (ibid.). Nonetheless, engineers in the United Kingdom are governed by a Royal Charter, passed in 1981 and revised in 2002.

This charter established categories of registration for engineers and the designations of "Chartered Engineer," "Incorporated Engineer," and "Engineering Technician." People seeking to acquire these titles must meet certain standards set by an umbrella organization recognized in the Charter, the EC, and be a member of a "licensed" engineering organization, such as the IEE.

BCS is a similar organization. Its 1984 Royal Charter recognized its right to examine and register computing professionals and grant a professional credential to members. As with engineers, practitioners need not join the BCS, but all those who meet the membership criteria and join are granted a recognized credential. Membership in the BCS (as in the IEE) requires meeting specified levels of professional competence as demonstrated by educational qualifications, examinations, and work experience. The IEE is an older and larger organization than the BCS—the former was established in 1871 and currently has 120,000 members (compared to the BCS's 47,000)—but in terms of professional status and governance, the two are similar. They have a similar regulatory status and similar levels of social closure. Their commonalities have been enhanced in the past 20 years as BCS has sought status as a licensed member organization of the EC. BCS succeeded in becoming a "nominating body" in 1985—able to nominate members for registration as chartered and technician engineers to the EC. In 1989, BCS became a full-chartered engineering institution, and in 1996, it became licensed by the EC (as one of its 35 member organizations). By the end of the 1990s, BCS had become one of the largest EC bodies, and over 8,000 of its members (at the time, 40 percent) were registered engineers (BCS 1999). Increasingly, BCS encourages an engineering focus for all Information Systems' practice and education (Finkelstein et al. 1993; Johnson 1997; BCS 2004).

Thus, in the United Kingdom, computing professionals and engineers have grown closer together, to the extent that many computing "professionals" are now engineers. This blending of the two professions has largely grown out of the cooperation between the dominant computing and engineering organizations toward the professionalization and certification of software engineers in the United Kingdom. Both the BCS and IEE believed that they could gain from such a collaboration: as Newton (1986:25) argued, cooperation made sense as "the BCS has the software engineers, and the IEE has the political awareness" to make professional progress a reality. Similarly, Conway et al. (1989:246) emphasized that this collaboration would bring together an institution with over a century of engineering expertise, with one "active in developing the earliest software education."

Overall, the emergence of SWE in the United Kingdom has been largely a collaborative project, and engineering and computing organizations have worked to claim this jurisdiction together. There is little evidence of interprofessional conflict. Their cooperation was fostered by similarities in organizational membership and strength, outlook, and in regulatory status. Each occupational group has a limited amount of social closure—they have restricted access to credentials but not to employment—and each can advance a legitimate claim to expertise in the field. The events in the United Kingdom are similar to those which Abbott describes as enclosure and amalgamation: a

SWE profession is growing out of two parent professions, computing and electrical engineering, and its emergence has brought the parent professions more closely together, encouraging computing's transformation into an engineering profession.

SWE IN CANADA

While ACM and BCS journals are replete with references to the software industry and software engineering, very little was published on these topics in CIPS's journals. The lack of attention to these subjects in CIPS's publications is revealing about the state of CIPS as an organization and the Canadian software industry. The industry was not well developed and while some felt that "Canada [was] uniquely poised to develop a world-class software industry," and that it had a great deal of competence in the field (Davies 1982:4), they expressed doubt over whether the industry could be "effective at taking advantage of the opportunity" (Potter 1982:5). The shortage of Canadian publications on SWE during this period is explained both by the lack of CIPS publications on any topic through most of the 1990s and the fact that its professional campaign to establish credentials for its members dominated the organization's attention in the 1980s and the early 1990s.

Despite the limited number of Canadian-based publications on SWE, Canadians have been interested in its development and have made contributions in this area. However, many have done so through their involvement in American-based organizations like the ACM and IEEE-CS. Canadian-based computing professionals have published on SWE in international journals and they have been heavily involved in the SWEBOK project. Their contributions do not demonstrate a distinctly Canadian viewpoint on the subject. It seems likely that the development of SWE as a field and occupation in Canada, during the 1980s and 1990s, resembled developments in the United States. Thus, while SWE as a set of techniques and practices allied with programming has been around for decades, SWE as a distinct occupation is a more recent phenomenon.[14] Its emergence in the Canadian context has been characterized by a great deal of interprofessional conflict between computing-related organizations and professional engineers.

Unlike their colleagues in the United Kingdom and the United States, Canadian engineers are self-regulating professionals governed by provincial legislation that protects the title of engineer—no one but a professional engineer can call him/herself an engineer—and enables engineers to establish strict criteria for entry to practice.[15] Currently, all professional engineers in Ontario, for instance, must graduate from a university engineering program accredited by the Canadian Engineering Accreditation Board, complete four years of relevant employment experience, and pass a professional practice examination on engineering law and ethics (Parnas 2002; Barker 2003). Those who complete all of these requirements are licensed as professional engineers and granted a "P. Eng" (professional engineer) designation. Given this legislation, anyone calling him-/herself a "software engineer," who is not a licensed engineer is, technically, in violation of the law.

As part of their own renewed professional project to increase the status and public relevance of engineering in Canada, Canadian engineers have consciously decided both to expand their involvement in the IT field, in particular by laying claim to the field of SWE, and to oppose and prosecute those illegally using the title "engineer" (CCPE 1997).[16] Engineering organizations are committed toward ensuring "that engineering in Canada is conducted by professional engineers" (p. 12). For Canadian engineers, SWE is not a computing discipline but is rightfully part of engineering. The commitment of Canadian engineers to claim and establish standards in the SWE field has disturbed many computer professionals who believe SWE is properly a computing specialty.

Until the late 1990s, there appears to have been little interprofessional conflict between computer professionals and engineers. Open conflict erupted in 1997 when the CCPE sued Memorial University in Newfoundland for attempting to offer a SWE degree through its computer science department; the engineers claimed infringement of copyright. They asserted that only accredited engineering faculties could offer courses in branches of engineering, although computer science departments across Canada had been offering courses in SWE since the 1970s. This case precipitated a great deal of discussion both within and between CIPS and engineering organizations, over which discipline could rightfully claim SWE and who could practice as software engineers.

The engineering position is straightforward: the terms "engineer" and "engineering" are protected titles owned by the engineering profession. Nonengineers cannot establish engineering programs, nor can they train people to become engineers. SWE is now a recognized engineering specialty: accredited SWE programs have been established in Canadian university engineering programs and the licensing of software engineers is underway. Opposing the efforts of engineers to claim the field for themselves are CIPS and other groups of computing workers who assert that "software engineering" is not an engineering discipline, but a term of historical origin that has come to be applied to a branch of practice within computer science and computing more broadly (Gabrini 2000; CIPS 2002a,b; Wordsworth 2002). CIPS defends the rights of nonengineers to perform SWE work and argues that it is IT professionals, and not engineers, who have "software systems expertise" and skills to perform the work safely (CIPS 2002a; Wordsworth 2002). Computing organizations reiterate that "the vast majority of Canada's academic software expertise resides" in computer science departments (CACS 2001) and stress that the effort to restrict SWE practice to engineers threatens the "right to practice" of many IT professionals. Of particular concern to many engineers is the fact that SWE is "increasingly being identified with the entire area of applied Computer Science" (CACS 2001). Some engineers claim that computer science is a theoretical discipline that informs SWE practice (Parnas 1998); from this perspective, there is no "applied" computer science, only engineering. Hence, the efforts of engineers to claim the jurisdiction of SWE threaten the boundaries of the discipline of computer science and the practice of many in the computing field.

Following its counterparts in the United States, CIPS has proposed to compromise with engineers and has advocated a joint accreditation committee that would see engineers and computer scientists working together to accredit, or at least to set accreditation criteria for, SWE programs in both computer science and engineering (Bassett 2002). A committee of CIPS and engineering professionals was established to explore joint accreditation, but it failed to reach a satisfactory solution, especially from engineering's point of view. As education is central to the licensing process in engineering, strict accreditation criteria are essential for maintaining professionalism in the field. Engineers have more rigid accreditation policies than do IT professionals—computer science is said to be a more flexible discipline in terms of options and specialization (Parnas 1998). It appears to be virtually impossible to establish accreditation criteria that would serve computer science, while preparing students adequately for an engineering license (Parnas 1998; DeVita 2000). Moreover, engineering leaders feel strongly that engineers must be trained in engineering faculties to acquire a broad engineering knowledge base and to preserve the "common engineering culture" (DeVita 2000). Furthermore, some argue, computer science graduates are not always "qualified to develop critical software products" (Parnas 2001:27); what the field needs are "engineers who understand basic engineering principles, but who also have the necessary specialized knowledge to develop software intensive products" (p. 39).

Thus, while organizations in the United Kingdom and the United States have managed to establish joint criteria for accreditation of SWE programs, such cooperation seems highly unlikely in the Canadian setting. As things stand currently, many Canadian computer science departments are at least scaling back on the use of the term SWE in the programs and courses they offer, while some others are joining with engineering faculties to offer SWE programs that meet engineers' requirements. SWE programs, accredited by an engineering body, are springing up in faculties of engineering across the country (Van Ihinger 2001). While some leaders have argued that there is a place in the software field for nonengineering software specialists (Parnas 2001; Parnas in Van Ihinger 2001), computer scientists in Canada have been slow to claim this domain as their own.

Why has interprofessional conflict over SWE been more prominent in Canada than in the United States and the United Kingdom? In this analysis, three variables seem particularly important. First, while there is a long history of cooperation between computing and engineering organizations in the United States and the United Kingdom, there is none in Canada. CIPS has neither formal nor informal ties with Canadian engineering organizations. While the ACM and IEEE-CS in the United States appear to have overlapping memberships, this is less true in Canada, where CIPS is primarily an organization of business-computing workers: for instance, in the early 1990s, 49 percent of CIPS members were managers and a further 25 percent were programmers and analysts. The presence of computing academics and engineers is quite small within CIPS, accounting for only 3 percent of members (CIPS 1990).

Second, computing workers in Canada are much less organized and less unified than are engineers. CIPS is a small organization representing about 6,000 IT workers, only a small fraction of those in the field, while engineering organizations claim to speak for all (160,000) licensed engineers. In the United States and the United Kingdom, the size and strength of computing and electrical engineering organizations are more comparable.

Third, and perhaps most important, is the difference in professional regulation between computing and engineering. As we have seen, computer professionals and engineers in the United Kingdom have a similar type of regulation and degree of professional status. The situation is not too different in the United States where most engineers and all computer professionals are not government-regulated or licensed. It is notable that the key area of interprofessional conflict to emerge in the United States is precisely in the area of licensing. In Canada, the gap between computer workers and engineers is the largest. Canadian engineers are licensed, highly educated, self-regulating professionals. Entry into the profession is limited through legislation, education, and licensing; engineers have a great deal of social closure. In contrast, computing workers in Canada are largely unregulated and only about half of them hold a university degree (Wolfson 2004). CIPS has established a credential for computing workers, but this credential is possessed by only some of its members and is only recognized by legislation in five of Canada's provinces.

To sum, while engineering is a prominent profession in Canada, computing possesses little professional status or social closure. Conflict is occurring as both groups of workers attempt to claim the jurisdiction of SWE. Given engineering's legislative and social status, it has nothing to gain from pursuing a compromise with computing organizations; engineering leaders have the strength and status to advance a claim to the jurisdiction on their own. Moreover, engineering and computing have "deep philosophical differences in professional approach" (CACS 2001:5) and do not have enough common ground for cooperation (also Parnas 2002). These differences in outlook and philosophy are somewhat evident in the United States as well, but are less apparent in the United Kingdom.

Given Canadian engineers' commitment toward extending professionalism within the SWE field and the fact that currently there are roughly 18,000 software engineers practicing in the country (according to the 2001 Canadian census), most of whom do not possess an engineering license, interprofessional conflict in this area should continue in Canada for some time.

CONCLUSION

In the sociological literature, Canada, the United States, and the United Kingdom have all been seen to follow a similar "Anglo-American model" of professionalization (Collins 1990; Evetts and Buchner-Jeziorska 2001). In some ways, this trend has continued with the rise of SWE: in all three nations, computing and engineering organizations have been at the forefront of an international movement to create a new occupation and profession. Nevertheless, significant international differences are evident. In particular, there has been evidence of interprofessional cooperation in the creation of SWE in the United Kingdom, evidence of both collaboration and conflict in the United States, and substantial interprofessional conflict in Canada. Together, these findings provide a challenge to Abbott's (1988:89) contention that it is interprofessional conflict that drives professional development and that with jurisdictional change, interprofessional contests are inevitable. Rather, they indicate that under certain circumstances, interprofessional cooperation and collaboration can shape professional development.

What factors shape interprofessional relations when two groups are interested in professionalizing the same jurisdiction? This analysis has focussed on the role of organization and legislative status in particular. The cases presented here suggest that when two groups with a claim to jurisdiction are similar in size, outlook, and legislative status—like the engineering and the computing professional organizations in the United Kingdom—cooperation is possible. In this country, both professional groups had an interest in claiming the jurisdiction, but neither had the status or influence to seize it for its own. Collaboration between the groups was encouraged by the government's concern with the lack of competitiveness and the lack of standards in the industry, and the common interests between the organizations. Only by working together could they attempt to claim jurisdiction and professionalize. In contrast, where professionalizing occupations have differing levels of power in terms of organizational size and regulatory and social status, as in the Canadian case, conflict along the lines discussed by Abbott is likely. Here, the more powerful occupational group attempts to dominate and seize the jurisdiction for itself. In Canada, the outcome of the interprofessional conflict will likely follow Abbott's (1988) general predictions: either engineers will succeed in dominating the field by subordinating other software workers or the field will be divided by task or client, with engineers taking on the most prestigious aspects.

The case of the United States, lying between these extremes, further illustrates the import of organization and legislative status in interprofessional relations. Collaboration and cooperation toward the creation of SWE as a distinct occupation and profession was encouraged by organizational similarities and ties between computing and engineering groups; however, the differences in legislative status between engineering and computing have provoked conflict. It is difficult to predict the course of future interprofessional relations in this field, but further conflict seems likely.

Ultimately, this analysis illustrates the role of power in shaping interprofessional relations. While Abbott (1988:140) downplayed the significance of power to jurisdictional disputes in the long run, this analysis suggests that power may be crucial in determining the nature and outcome of interprofessional relations. Future research should continue to explore the role of power in shaping both professionalization and interprofessional relations.

Future research on developments in SWE in these nations should also explore many other factors not discussed here, including differing national traditions in and attitudes toward professional regulation, occupational cultures, geography, and public and employer opinions. Employers will play a particularly important role in determining the outcome of professional projects in SWE. Currently, people without credentials work in the field in all three nations and if neither employers nor governments insist on credentials, professional leaders' efforts to restrict entry to practice (to achieve more social closure) and raise the status of SWE as a profession will be unsuccessful. Future research must also consider the impact of trends occurring on an international level. The internationalization of software development, international professional activity, and international employment practices have implications for the professional development of SWE in many nations. For instance, the standards established by national professional organizations and governments may be circumvented by employers who hire offshore workers employed in regions with different standards. Moreover, international professional activities and interprofessional relations may challenge or support efforts at the national level. In the end, the professionalization of SWE will be shaped not only by interprofessional relations at the national level, but interprofessional relations at the international level, as well as many other national and international trends.

ACKNOWLEDGMENTS

I would like to thank Erin Demaiter and Rob Downie for their research assistance, and Julie McMullin, Peter Kivisto, and the anonymous reviewers for their comments on earlier drafts of this article. This research was supported by a grant from the Social Sciences and Humanities Research Council of Canada.

NOTES

[1]By legislative status, I mean whether the professions are state regulated, as well as the nature and the extent of professional privileges that state legislation grants them. Regulatory status is important because it is through state regulation that professions come to acquire power (Freidson 2001:128).

[2]Abbott (1988) did not specify how conflict among professions might differ from conflict among other organized "expert" occupations. Given this ambiguity, I have decided to use the term "interprofessional conflict" when discussing relationships among software engineering, computing and engineering, even though the claims of each group to professional status may be debated in some contexts.

[3]That interprofessional cooperation occurs is, at times, implied in Abbott's model, but the incidences and contexts of cooperation are not explored.

[4]Article topics included the following: president's letters, editorials, discussions of education programs, curriculum development, credentials, accreditation, organizational histories, organizational change, working experiences, occupations, and the careers of computing workers, the impact of technology and computing practice on society, public perceptions of computing, ethics, working relations, the state of the computing/IT profession and computing knowledge, professional membership, government relations, and legal issues.

[5]Approximately, 83 articles were taken from BCS journals, 125 (plus 17 Web publications) from CIPS journals, and 147 articles and 210 letters from the *Communications of the ACM*.

[6]After this date, the journal was succeeded by the *IEE Transactions on Software Engineering* (published jointly with the BCS). With the change, the journal became more technical in focus, and hence, no longer carried many articles related to professional issues. The BCS's more technical *Computer Journal* was also perused to glean additional material, especially for the years 1989 to 1994 for which it proved too difficult to access the *Computer Bulletin*.

[7]CIPS was largely silent on software engineering as a discipline and occupation until the late 1990s when Canadian engineers claimed software engineering education as solely their domain.

At this time, CIPS began to advance its own claims to the field. CIPS's writings on SWE, then, are largely a product of interprofessional conflict, and have been generated in response to claims advanced by engineers.

[8]Computer groups were formed within the American Institute of Electric Engineers and the Institute of Radio Engineers in the 1940s. In the early 1960s, these two organizations merged to form the Institute of Electrical and Electronic Engineers (IEEE); at this time, their computer groups merged to form the IEEE-CS.

[9]These international differences in professional activity in computing reflect somewhat differing approaches to professionalism in the three countries more generally. In particular, the United States has been historically more opposed to monopolies and restrictions on practice than the United Kingdom or Canada.

[10]As Kraft (1977) argues, programming has been rationalized and routinized, prompting some deskilling. This routinization itself may have encouraged skilled software workers to embrace a new title for their work.

[11]For instance, the two had joined together, with other societies, to form American Federation of Information Processing Societies (AFIPS), the body that represented computing workers internationally. They had also jointly organized conferences for decades, had been two of the founding organizations of ICCP, a body providing credentials to IT workers upon examination, and had joined together to form the CSAB to accredit university computer science programs.

[12]The development of the SWEBOK was led by a team at the University of Quebec at Montreal, but involved consultation with thousands of software practitioners in 42 countries (Abran et al. 2004).

[13]Here, the change in accreditation policies is significant. CSAB's current association with ABET should ensure that computer science SWE programs meet engineering license requirements.

[14]It is only recently that Canadian occupational classifications have a category for SWE.

[15]Unlike in the United States, licensing is mandatory for engineers in Canada.

[16]In this endeavor, they have had limited success. Engineers in Quebec recently won a court battle against Microsoft for the latter's illegal use of the term engineer in its certification program; however, the battle was at best a moral one, as the penalty was only a $1,000 fine (Schick 2004). A recent Alberta case against an Apple Canada-certified systems engineer was dismissed by the court when it deemed the defendant "presented no injury to the public" (Aschaiek 2004:15).

REFERENCES

Abbott, Andrew. 1988. *The System of Professions*. Chicago, IL: University of Chicago Press.

Abran, Alain, Pierre Bourque, and Leonard L. Tripp. 2004. *Guide to the Software Engineering Body of Knowledge, Final Version*. IEEE Computer Society. Retrieved May 19, 2005 (http://www.swebok.org).

Abran, Alain, James W. Moore, Pierre Bourque, and Robert Dupuis, eds. 2001. *Guide to the Software Engineering Body of Knowledge, SWEBOK*. IEEE Computer Society. Retrieved June 23, 2003 (http://www.swebok.org).

Association for Computing Machinery (ACM). 2000. "A Summary of the ACM Position on SWE as a Licensed Profession." Retrieved April 6, 2004 (http://www.acm.org/serving/se_policy/selep_main.html).

——. 2001."On Licensing of Software Engineers Working on Safety-Critical Software: Final report of the ACM Task Force." Retrieved April 16, 2004 (http://www.acm.org/serving/se_policy/selep_main.html).

——. 2005. *Communications of the ACM*. Retrieved May 19, 2005 (http://portal.acm.org/browse_dl.cfm?linked=1&part=magazine&idx=J79&coll=ACM&dl=ACM&CFID=23492031&CFTOKEN=97926218).

Adams, Tracey L. 2004a. "Inter-Professional Conflict and Professionalization: Dentistry and Dental Hygiene in Ontario." *Social Science and Medicine* 58:2243–52.

——. 2004b. "Professionalization in Computing Related Occupations: Canada, the U.S. and Britain." WANE Working Paper, February 2004. Retrieved June 25, 2005 (http://www.wane.ca).

Andriole, Stephen J. and Peter A. Freeman. 1993. "Software Systems Engineering: The Case for a New Discipline." *Software Engineering Journal* 8:165–79.

Aschaiek, Sharon. 2004. "Alberta Case Thrown Out: Debate Continues on Software Titles." *Engineering Dimensions* (March/April):15.

Bagert, Donald J. 1999. "Viewpoint: Taking the Lead in Licensing Software Engineers." *Communications of the ACM* 42(4):27–9.

Barker, Roger F. (P. Eng and Registrar, PEO). 2003. Letter re Eng licensure standards. Retrieved April 8, 2004 (http://www.peo.on.ca/registrar/Barker_ITBusiness.html).

Barron, D. W. 1975. "Professional Programmers?" *Computer Bulletin* 2:5.

Bassett, Paul. 2002. "Bassett Proposal to Restart SEAB Discussion." Letter dated April 3, 2002. Retrieved April 29, 2003 (http://www.cs.usask.ca/spec_int/cacs/page13.htm).

BCS. 1999. "News in Brief: Engineering Growth." *The Computer Bulletin* (May):10.

——. 2004. "Projects Report: Time for National Action." *The Computer Bulletin* (July): 23.

BCS/IEE. 1989. *A Report on Undergraduate Curricula for Software Engineering*. London: Institution of Electrical Engineers.

Bechtolsheim, Stephan V. 1985. "Letter: On the Potential ACM/IEEE-CS Merger." *Communications of the ACM* 28:237–8.

Berztiss, Alfs. 1987. "A Mathematically Focussed Curriculum for Computer Science." *Communications of the ACM* 30:356–65.

Bott, Frank. 1989. "Software Engineering Education." *Software Engineering Journal* 4:174–5.

Bourque, Pierre, Robert Dupuis, Alain Abran, James W. Moore, Karen Shyna, Leonard Tripp, Bryan Pflug, Marcela Maya, and Guy Tremblay. 1998. *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society. Retrieved June 23, 2003 (http://www.swebok.org).

Brown, P. J. 1985. "Software Engineering . . . and Documentation." *Computer Bulletin*, Series III 1:16–7.

Bureau of Labor Statistics. 2001. "Fastest Growing Jobs 2000–2010." Retrieved August 4, 2004 (http://www.bls.gov/opub/ted/2001/dec/wk1/art02.htm).

Canadian Association of Computer Science (CACS). 2001. "CACS/AIC position on SEAB." Retrieved April 6, 2004 (http://www.cs.usask.ca/spec_int/cacs/page14.htm).

Canadian Council of Professional Engineers (CCPE). 1997. *A Vision for the Engineering Profession in Canada, CCPE*. Retrieved December 3, 2003 (http://www.ccpe.ca).

de Champeaux, Dennis. 2002. "Software Engineering Considered Harmful." *Communications of the ACM* 45:102–4.

Chapman, Gary. 1990. "Bugs in the Program." *Communications of the ACM* 33:251–2.

CIPS. 1969. "Report by the Accreditation Committee to the Directors of the Information Processing Society of Canada." *IPSOC Quarterly Bulletin* 9(Spring supplement):1–12.

——. 1973. "Students, Businessmen and Academics Question the Content of Computer Science Education." *CIPS Computer Magazine* 4:4–6.

——. 1990. "Who Have We Attracted? Members by Job Category." *CIPS National Newsletter* 2:2.

——. 2002a. "Software Engineering: Position Paper." Retrieved April 29, 2003 (http://www.cips.ca/it/position/softeng/).

——. 2002b. "What's in a Name? Tech Sector Battles Engineers on 'Software Engineering.' " Retrieved April 29, 2003 (http://www.cips.ca/news/media/news.asp?aID=1387).

Coburn, David. 1994. "Professionalization and Proletarianization: Medicine, Nursing and Chiropractic in Historical Perspective." *Labour/Le Travail* 34:139–62.

——. 1999. "Phases of Capitalism, Welfare States, Medical Dominance, and Health Care in Ontario." *International Journal of Health Services* 29:833–51.

Collins, Randall. 1990. "Changing Conceptions In the Sociology of Professions." Pp. 11–24 in *The Formation of Professions*, edited by Rolf Torstendahl and Michael Burrage. London: Sage.

Collins, W. Robert, Keith W. Miller, Bethany J. Spielman, and Phillip Wherry. 1994. "How Good Is Good Enough: An Ethical Analysis of Software Construction and Its Use." *Communications of the ACM* 37:81–91.

Conway, Don E., S. C. Dunn, and Gillian S. Hooper. 1989. "BCS and IEE Accreditation of Software Engineering Courses." *Software Engineering Journal* 4:245–8.

Conway, Don and Gillian Hooper. 1988. "Accreditation of Courses." *Computer Bulletin*, Series III 4:4–7.

Crinean, Kay and Lisa Baguley. 1986. "The Starts Initiative." *Computer Bulletin*, Series III 2:35, 37.

Computer Science Accreditation Board (CSAB). 2004. Retrieved April 16, 2004 (http://www.csab.org).

Davies, Jack. 1982. "Opportunities Are There—If We Don't 'Muff' Them." *CIPS Review* 6(J/A):4.

DeMarco, Tom. 1999. "ACM Forum: Certification or Decertification?" *Communications of the ACM* 42(7):11–13.

Denning, Peter J. 1990. "Human Error and the Search for Blame." *Communications of the ACM* 33:6–7.

——. 1998. "Computer Science and Software Engineering: Filing for Divorce?" *Communications of the ACM* 40:128.

Denning, Peter J., Douglas E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, and Paul R. Young. 1989. "Computing as a Discipline." *Communications of the ACM* 32:9–23.

DeVita, Peter. 2000. "Memorandum: Recommendations of the (PEO) Software Engineering Panel." (http://www.peo.on.ca/news/Software_ccpesubmission.htm).

EC Engineering Council, U.K.). 2003. "Registration." Retrieved December 3, 2003 (http://www.engc.org.uk/registration/engineerprotect.asp).

Ensmenger, Nathan. 2001. "The 'Question of Professionalism' in the Computer Fields." *IEEE Annals of the History of Computing* 23:56–74.

Evetts, Julia and Anna Buchner-Jeziorska. 2001. "The Professionalization of Knowledge in European Markets: Engineering in the U.K. and Poland." *Current Sociology* 49:133–47.

Finkelstein, Anthony, Jeff Kramer, Samson Abrams, Krysia Broda, Sophia Drossopoulou, and Susan Eisenbach. 1993. "An Integrated Engineering Study Scheme in Computing." *The Computer Journal* 36(4):320–34.

Frailey, Dennis J. 1988. "More on the Computer Science Decline." *Communications of the ACM* 31(8):946–7.

——. 1999. "Viewpoint: Licensing Software Engineers." *Communications of the ACM* 42(12):29–30.

Freidson, Eliot. 1994. "The Futures of Professionalization." Pp. 106–27 in *Professionalism Reborn*. Chicago, IL: University of Chicago Press.

——. 2001. *Professionalism: The Third Logic*. Chicago, IL: University of Chicago Press.

Gabrini, Philippe. 2003. "Software Engineering: Where Does It Belong?" Retrieved April 29, 2003 (http://www.cips.ca/it/position/software/).

Gries, David. 1991. "Teaching Calculation and Discrimination: A More Effective Curriculum." *Communications of the ACM* 34:45.

Gotterbarn, Don, Keith Miller, and Simon Rogerson. 1997. "Software Engineering Code of Ethics." *Communications of the ACM* 40:110–18.

Grosch, Herb. 1987. "Letters: After Nineteen Years." *Communications of the ACM* 30:996.

Hafferty, Frederic and Donald Light. 1995. "Professional Dynamics and the Changing Nature of Medical Work." *Journal of Health and Social Behaviour* 35:132–53.

Hartley, Heather. 2002. "The System of Alignments Challenging Physician Professional Dominance." *Sociology of Health and Illness* 24:178–207.

Hoare, C.A.R. 1975. "Software engineering." *Computer Bulletin,* Series II 6:6–7.

——. 1981. "Professionalism." *Computer Bulletin*, Series II, 29:2–4.

Hoffnagle, Gene F. 1991. "Epilogue: Engineering the Software Development Process." *Software Engineering Journal* 6:inside print cover.

Johnson, Luanne (James). 1998. "A View from the 1960s: How the Software Industry Began." *IEEE Annals of the History of Computing* 20:36–42.

Johnson, Roger. 1997. "What Are We Here For?" *Computer Bulletin* (October):14–15.

Kelner, Merrijoy, Heather Boon, Beverly Wellman, and Sandy Welsh. 2004. "Responses of Established Healthcare to the Professionalization of Complementary and Alternative Medicine in Ontario." *Social Science and Medicine* 59:915–30.

Knight, John and Nancy Leveson. 2002. "Should Software Engineers be Licensed?" *Communications of the ACM* 45:87–90.

Kocher, Bryan. 1989. "President's Letter." *Communications of the ACM* 32:660–1.

——. 1990. "President's Letter: Eschew Obfuscation." *Communications of the ACM* 33:625.

Kraft, Philip. 1977. *Programmers and Managers*. New York: Springer-Verlag.

Larson, Magali Sarfatti. 1977. *The Rise of Professionalism*. Berkeley: University of California Press.

Lehman, M. M. 1991. "Software Engineering, the Software Process and Their Support." *Software Engineering Journal* 6:243–4.

Leicht, Kevin T. and Mary L. Fennell. 2001. *Professional Work*. Malden, MA: Blackwell.

Leveille, John J. 2002. "Jurisdictional Competition and the Psycho-Analytic Dominance of American Psychiatry." *Journal of Historical Sociology* 15:252–80.

McCracken, Daniel D. 1987. "Ruminations on Computer Science Curricula." *Communications of the ACM* 30:3–5.

McIlroy, M. D. 1990. "Letters: Green Light for Bad Software." *Communications of the ACM* 33:479.

Murphy, Raymond. 1988. *Social Closure*. Toronto, Canada: Oxford University Press.

Naur, P. and Randell, B. 1968. "Software engineering." Report on a conference sponsored by the NATO Science Committee, Garmisch Germany, October 1968.

Newton, R. W. 1986. "Software, a Vital Key to U.K. Competitiveness." *Computer Bulletin*, III 2:24–28.

Norris, Pauline. 2001. "How Are 'We' Different from 'Them?' " *Sociology of Health and Illness* 23:824–43.

Notkin, David, Michael Gorlick, and Mary Shaw. 2000. *An Assessment of Software Engineering Body of Knowledge Efforts*. A Report to the ACM Council. Retrieved May 2000 (http://www.acm.org).

Orden, Alex. 1967. "The Emergence of a Profession." *Communications of the ACM* 10:145–7.
Ould, Martyn and David Thewlis. 1987. "Making Software Engineering Standards Usable." *Computer Bulletin*, III, 3:28–30.
Parnas, David L. 1998. "Software Engineering Programmes Are Not Computer Science Programmes." *Annals of Software Engineering* 6:19–37.
——. 2001. "Why Software Developers Should Be Licensed." *Engineering Dimensions* (May/June):36–39.
Parnas, David L. 2002. "Licensing Software Engineers in Canada." *Communications of the ACM* 45:96–98.
Potter, Mike. 1982. "Penetrating U.S. Market Demands Careful Strategy." *CIPS Review* 6(July/August):5–7.
Pyle, I. C. 1986. "Software Engineers and the IEE." *Software Engineering Journal* 1:66–68.
Rada, Roy. 1999. "Sharing Standards: IT Skills Standards." *Communications of the ACM* 42(4):21–26.
Ralston, Anthony. 1984. "The First Course in Computer Science Needs a Mathematics Co-Requisite." *Communications of the ACM* 27:1002–5.
Rickert, Neil W. 1989. "Letters: Computing as a Discipline." *Communications of the ACM* 32:1286–7.
Schick, Shane. 2004. "Quebec Engineers Win Court Battle against Microsoft." Retrieved April 8, 2004 (http://www.itbusiness.ca).
Shore, John. 1988. "Why I Never Met a Programmer I Could Trust." *Communications of the ACM* 31:372–5.
Starr, Paul. 1982. *The Social Transformation of American Medicine*. New York: Basic Books.
Thomas, Martyn. 1987. "Professionalism and Software Engineering." *Computer Bulletin*, Series III, 3:3.
Timmermans, Stefan. 2002. "The Cause of Death vs. the Gift of Life: Boundary Maintenance and the Politics of Expertise in Death Investigation." *Sociology of Health and Illness* 24:550–74.
Van Ihinger, Sharon. 2001. "Debate Continues on Joint Accreditation Council: CEAB Accredits First Software Programs." *Engineering Dimensions* (July/Aug):8–9.
Wagner, F. 1990. "Letters." *Communications of the ACM* 33:628–9.
Wilkes, M. V. 1991. "Software and the Programmer." *Communications of the ACM* 34:23–4.
Willis, Evan. 1983. *Medical Dominance*. Sydney, Australia: George Allen and Unwin.
Winder, Russel, Robert Cole, and Charles Easteal. 1987. "Software Engineering in a First Degree." *Software Engineering Journal* 2:133–9.
Witz, Anne. 1992. *Professions and Patriarchy*. London: Routledge.
Wolfson, William C. 2004. *Analysis of Labour Force Survey Data for the Information Technology Occupations, 2000–2003*. Software Human Resource Council. Retrieved June 25, 2005 (http://www.shrc.ca).
Wordsworth, Charles. 2002. "Letters to the Editor—Engineering Moniker—the Sequel." *Computing Canada* 28(August 23). Retrieved April 14, 2004 (http://www.peo.ca).