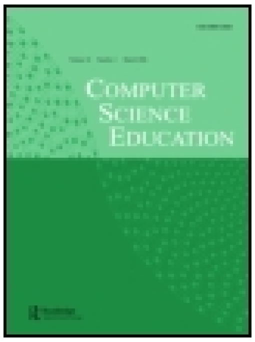

Computer Science Education

ISSN: 0899-3408 (Print) 1744-5175 (Online) Journal homepage: https://www.tandfonline.com/loi/ncse20

Software Engineering as a Professional Discipline

Donald Bagert & Nancy R. Mead

To cite this article: Donald Bagert & Nancy R. Mead (2001) Software Engineering as a Professional Discipline, Computer Science Education, 11:1, 73-87, DOI: 10.1076/csed.11.1.73.3841

To link to this article: https://doi.org/10.1076/csed.11.1.73.3841

Computer Science Education
2001, Vol. 11, No. 1, pp. 73–87

0899-3408/01/1101-0073$16.00

Software Engineering as a Professional Discipline

Donald Bagert
Texas Tech University

Nancy R. Mead
Software Engineering Institute

ABSTRACT

In this paper we discuss some of the aspects of software engineering as a professional discipline. After a brief discussion of software engineering as a profession, we focus on accreditation, licensing, and certification. We examine historic trends, current status, and the future outlook in each of these areas. We anticipate that this will continue to be an area that sparks professional interest and debate, as the field of software engineering matures.

INTRODUCTION

Software engineering has become increasingly recognized as a professional discipline distinct from both computer science and from other engineering fields. Many professional disciplines have the following components:

- a distinct body of knowledge (BoK),
- a code of ethics and professional practices,
- educational programs and curriculum models for people desiring to become professionals in that discipline,
- a means for assessing educational programs (accreditation), and
- a means of identifying competent professionals in the discipline (licensing and certification).

This paper will look at the evolution and current status in some countries of accreditation for software engineering degree programs as well as the licensing and certification of software engineers. This paper will also provide

Correspondence: Donald Bagert, Department of Computer Science, Texas Tech University, MS3 104, Lubbock, TX 79409, USA. Tel: +1 806 742 1189. E-mail: don.bagert@ttu.edu

some insights about the future of software engineering accreditation, licensing, and certification.

BACKGROUND

There is often confusion, or at least uncertainty, about accreditation, certification, and licensing as they relate to software engineering and software engineering education (Mead, 1997). The differences in meaning among these terms, the processes involved, and the implications for software engineering education are sometimes unknown or misunderstood.

The terms *accreditation*, *certification*, and *licensing* are used in various ways, sometimes interchangeably and sometimes overlapping. The definitions below are provided in order to establish a framework for discussion of issues related to these processes (Mead, 1998).

Accreditation

Accreditation applies to educational programs (as opposed to individuals). The two most common kinds of accreditation are

- institutional, which applies to an educational institution as a whole; and
- programmatic or specialized, which applies to individual degree programs or academic units. These are usually professionally oriented baccalaureate programs with a primary goal of preparing their graduates to function in a professional practice.

Accreditation is usually performed by organizations in conjunction with professional societies. In some countries, there are organizations that evaluate and accredit the accrediting agencies themselves, for example, the Commission on Recognition of Postsecondary Accreditation (CORPA) in the United States.

Certification and Licensing

Certification applies to individuals and usually is intended to document the achievement of some level of skill or capability. It is normally a voluntary process that is administered by a profession (Ford, 1996). This type of certification can be given through a professional society or an independent organization. Also, some companies offer programs that certify that individuals have particular skills regarding their specific products. In many cases, individuals are eligible for certification regardless of the country in which they practice or reside.

Licensing also applies to individuals and is a more formal version of certification that involves a government-sanctioned or government-specified process. In most countries, this is done at the national level; in the United States it is done at the state level and in Canada at the province level. What professions are licensed varies from country to country, as do the conditions under which a license is required to practice in a field.

In some cases, boundaries between licensing and certification are blurred. For instance, in the United Kingdom the Institution of Electrical Engineers (IEE)[1] certifies electrical engineers as ‘‘chartered engineers’’ (in conjunction with the Engineering Council[2]) under a royal charter from the British government.

Certification and licensing have been performed in other professions for years. For example, Ford discusses professional certification programs for accounting, medicine, and engineering, and Shaw makes reference to the medical profession (Ford, 1996; Shaw, 1996). Certification in these areas is often used as a condition of employment or as a requirement for performing certain functions in the profession. Medicine and law are the licensed professions most widely known to the public; for instance, in the US all states require that medical doctors, dentists, and lawyers be licensed in order to practice the profession without supervision.

Engineering is a field where licensing is commonplace in many countries. However, which engineers must be licensed varies from country to country and in the United States varies from state to state. Most engineers are not licensed, and licensing is most often required for safety-critical work such as that performed by many civil engineers.

Engineers may be licensed, but the requirement and expectation for licensing varies widely among different engineering disciplines. To illustrate, the percentage of engineers in common engineering disciplines who seek licenses in the United States is shown in Table 1.

Relationship Between Accreditation and Licensing/Certification

There is often a relationship between accreditation and licensing or certification. For instance, in the US, some states require a student to have graduated with an undergraduate degree accredited by the Accreditation Board of Engineering and Technology (ABET)[3]. In turn, the National Council

1. http://www.iee.org.uk
2. http://www.engc.org.uk
3. http://www.abet.org

Table 1. Percentage of Licensed Engineering Graduates (Ford, 1996).

Discipline	Licensed (%)
Civil	44
Mechanical	23
Electrical	9
Chemical	8
All engineers	18

of Examiners for Engineering and Surveying (NCEES)[4], which creates engineering licensing exams for most of the United States, will not offer examination for engineering disciplines that do not have at least one accredited program in that area. The implication is that a student graduating from an accredited engineering program in the United States should have the educational background sufficient for passing the appropriate licensing exams given to new graduates.

Development of Software Engineering as a Profession

In some countries (such as the United Kingdom and Australia) the development of software engineering as a profession, including licensing and society certification, has been ongoing for many years. However, the first multinational effort of note was launched by the Association for Computing Machinery (ACM) and the Computer Society of the Institute of Electrical and Electronics Engineers (IEEE-CS) in 1993 by their establishment of an ad hoc Joint Steering Committee for the Establishment of Software Engineering as a Profession[5]. The initial recommendations of the steering committee were to define ethical standards and to determine the required body of knowledge and recommended practices in software engineering, and the appropriate curricula to acquire the body of knowledge. Three task forces, one for each area, were established. The ethics task force developed the *Software Engineering Code of Ethics and Professional Practices*, which was eventually adopted by both societies (Gotterbarn, Miller, & Rogerson, 1999). The BoK task force did a pilot for a survey intended to collect information from practitioners in order to determine the software engineering BoK. Finally, the education task force

4. http://www.ncees.org
5. http://www.computer.org/tab/seprof/

developed model accreditation criteria for undergraduate software engineering programs. This model criteria, which was also intended as a requirements specification for a software engineering undergraduate curriculum model, will be discussed further in Section 3.

After the pilot survey was done by the BoK task force, the joint steering committee realized that it would take more than a group entirely composed of volunteers to define the BoK for software engineering. To that end, in 1997 IEEE-CS created and provided the initial funding for the Software Engineering Body of Knowledge (SWEBOK) project, headquartered at the Université du Québec à Montréal (Dupuis & Bourque, 1999).

It was at about this time that ACM and IEEE-CS realized that a permanent entity should replace the joint steering committee and sponsor and supervise projects such as SWEBOK. This led to the creation in 1998 of the Software Engineering Coordinating Committee (SWECC, or SWEcc), which officially took the place of the joint steering committee on 1 January 1999 (Bagert, 1998b). SWECC took on the sponsorship of three ongoing projects: the code of ethics (which was in its final stages of approval by the societies); SWEBOK; and the curriculum model project, which was renamed SWEEP (Software Engineering Education Project).

No activities of the joint steering committee or SWECC were related to licensing. However, due to the nature of the SWECC-sponsored projects, there were still some concerns regarding how the committee's work would affect licensing. This will be discussed more in Section 4.

Impetus for Software Engineering Accreditation and Licensing

When undergraduate degree programs in software engineering are developed in a particular country, there is a desire for that country's engineering accreditation body to develop criteria for accrediting such programs, as it would do for other engineering disciplines. Also, because governmental bodies in some countries restrict the use of the term "engineer" to those people who are licensed or certified—and the job title of "software engineer" is becoming more and more commonplace—there is interest in licensing software engineers through the engineering licensing boards.

SOFTWARE ENGINEERING ACCREDITATION

Most accreditation actions are made on the basis of a determination as to whether an institution or program satisfies a published set of criteria. Usually the institution or program must submit a written self-study representing its analysis of how it satisfies the criteria as well as other information that might be relevant to an accreditation decision. Later an evaluation team from the accreditation agency visits the institution and makes a report to the accreditation agency on its findings. It might also make a recommendation concerning accreditation.

The final accreditation actions are determined by a committee or other group from the accreditation agency. The entire process usually takes from one to two years, not counting the time required to determine whether a program is ready to seek accreditation and to make any changes needed to satisfy the accreditation criteria. Accreditation is granted for some period of time after which the process must be repeated for reaccreditation.

United States

Historically, accreditation of engineering programs in the United States has been conducted by the Engineering Accreditation Commission (EAC) of the ABET and accreditation of computer science programs has been conducted by the Computer Science Accreditation Commission (CSAC) of the Computer Science Accreditation Board (CSAB[6]), respectively. The process for both agencies is the one described above, and accreditation is usually granted for a period of six years.

IEEE has developed criteria for the accreditation of software engineering programs by the EAC of ABET (Gillespie, 1997). ABET and CSAB have recently decided to merge, with the CSAB becoming a participating body of ABET. Under the plan, CSAB will become a member of ABET and CSAC will be replaced by a new ABET commission, the Computing Accreditation Commission (CAC). CSAB will have lead responsibility for the accreditation of software engineering programs by the EAC, as well as lead responsibility for the accreditation of computer science programs by the CAC.

ABET has announced its intention to accredit software engineering programs. Such accreditation would be carried out by the EAC, which already has criteria for software engineering. Starting in fall 2001, evaluators for software engineering will be selected from a list of eligible persons provided to the EAC by CSAB. CSAB will have responsibility for software engineering criteria. It is expected that they will remain similar to all other EAC criteria.

6. http://www.csab.org

Once CSAB becomes a participating body of ABET, it will examine the existing software engineering criteria.

In September 1999, ABET announced the appointment of Doris K. Lidtke, a past CSAB President, as the Adjunct Accreditation Director for Computing. Her primary focus will be to assist in the integration of CSAB/CSAC into ABET, and to provide a bridge between the communities. During the transition, Lidtke will provide guidance to those institutions with programs seeking initial or continuing accreditation with CSAC. She will also assist in the formation of the new ABET Computing Accreditation Commission and will coordinate implementation of the new CSAC criteria for evaluating computing programs.

Although no software engineering program has requested evaluation from ABET in fall 2000, it is almost certain that one or more programs will be visited in fall 2001. It is anticipated that the software engineering evaluators in fall 2001 will be recommended by CSAB. At present there are a number of software engineering degree programs in the United States. It is likely that the Rochester Institute of Technology will be the first school to request accreditation of an undergraduate degree program in software engineering.

The accreditation criteria include general criteria, which each program must satisfy, and program criteria, which apply to the faculty and curriculum of a specific program. In software engineering, the faculty criteria are inherited from the general criteria, except as noted below. The specific criteria for software engineering are as follows:

> Curriculum: The curriculum must provide both breadth and depth across the range of engineering and computer science topics implied by the title and objectives of the program. The program must demonstrate that graduates have the ability to analyze, design, verify, validate, implement, apply, and maintain software systems; the ability to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and supporting disciplines to complex software systems; and the ability to work in one or more significant application domains.
>
> Faculty: The program shall demonstrate that those faculty teaching core software engineering material have practical software engineering experience.

US institutions that offer undergraduate software engineering degree programs as of fall 2000 include Auburn University, Milwaukee School of Engineering, Mississippi State University, Monmouth University, Rochester

Institute of Technology, University of Michigan, Dearborn, and University of Wisconsin, Platteville.

United Kingdom

Undergraduate degree programs in software engineering have been slowly developed in most countries. One exception is the United Kingdom, where the British Computer Society (BCS)[7] and the Institution of Electrical Engineers (IEE) have worked together to promote software engineering as a discipline. The first major effort to develop a software engineering curriculum model was a joint effort by these two societies (BCS/IEE, 1989). Edwards (1999) of the University of Sunderland stated that in the UK as of mid-1999 there were 26 bachelor's degree programs in software engineering or "in computing with a strongly themed software engineering stream."

IEE, in cooperation with BCS, has accredited software engineering programs in the United Kingdom for a number of years. Currently, about 15 schools with degree courses (programs) that have the words "software engineering" in their titles are accredited by IEE (2000). According to the IEE web page, the main features IEE uses in considering a course for accreditation are

- quality of students (entry standards, motivation, etc.),
- quality of staff (qualifications, research, publications),
- evidence of quality assurance procedures,
- aims and philosophy of the course,
- structure and content of the course,
- inclusion of design,
- inclusion of engineering applications through case studies and projects,
- assessment methods,
- industrial contact,
- resources.

Canada

Canada had a legal dispute over the use of the term "engineering" by software engineers and in universities (Peters, 1999). The Association of Professional Engineers and Geoscientists of Newfoundland (APEGN) and the Canadian Council of Professional Engineers (CCPE) filed a statement of claim alleging

7. http://www.bcs.org.uk

trademark violation by Memorial University of Newfoundland (MUN) for using the name "software engineering" for a baccalaureate program. The APEGN and the CCPE claimed the program was not accreditable as an engineering program.

Subsequently, the parties came to an agreement: MUN dropped the use of the title "software engineering" for its program, APEGN and CCPE discontinued their lawsuit, and the three organizations agreed to work together to define the appropriate uses of the term "software engineering" within Canadian universities (AUCC/CCPE, 1999).

As a result of this action, it is expected that the Canadian Engineering Accreditation Board (CEAB) will soon begin to accredit software engineering undergraduate degree programs. As of fall 2000, Canadian bachelor's degree programs in software engineering exist at Concordia University, McMaster University, the University of Ottawa, and the University of Western Ontario.

Below is a summary of the curriculum content required for engineering accreditation by CEAB. (AU stands for Accreditation Units. Fifteen AU is approximately equivalent to one semester-hour in a US institution.)

- Mathematics: A minimum of 195 accreditation units (AU) of mathematics including appropriate elements of linear algebra, differential and integral calculus, differential equations, probability, statistics, and numerical analysis.
- Basic sciences: A minimum of 225 AU of basic (natural) sciences, defined as elements of physics and chemistry and, where appropriate, elements of life sciences and earth sciences.
- Engineering sciences and engineering design: A minimum of 900 AU of a combination of engineering sciences and engineering design. Within this combination, neither the engineering sciences nor the engineering design portion can be less than 225 AU. The engineering curriculum must culminate in a significant design experience.
- Complementary studies: A minimum of 225 AU of studies in humanities, social sciences, arts, management, engineering economics, and communication that complement the technical content of the curriculum.

The entire program must include a minimum of 1,800 AU. Also, appropriate laboratory experience must be an integral part of the course, and the role and responsibilities of the professional engineer in society must also be covered.

In 1999, the Canadian Computer Science Accreditation Council drafted and considered accreditation guidelines for computer science and software engineering (Gabrini, 1999).

Australia

According to Grant (2000) of the Swinburne University of Technology, bachelor's degree programs in software engineering are being offered by at least nine of Australia's 39 universities, with more being planned. The Institution of Engineers, Australia (IEAust)[8] has been accrediting software engineering programs since 1993. Currently, software engineering is accredited at the University of Melbourne and Murdoch University.

Accreditation in Other Countries

In some other countries, engineering as a discipline is accredited, and there is no separate accreditation for specialities, and therefore no difficulties arise when software engineering degrees are added. In much of Europe, accredited programs often have departments and degrees in computing, so again software engineering exists within that discipline rather than as a separate degree.

LICENSING

United States

In most states, applicants for a professional engineer (PE) license have taken and passed two examinations given by NCEES. The first test is the Fundamentals of Engineering (FE) examination given to people who are about to graduate or have recently graduated with a bachelor's degree. (Some states require that the degree be from an ABET-accredited program, while other states allow degrees from "related programs" such as computer science.) The morning part of the FE exam is taken by all applicants and includes mathematics (calculus, differential equations, linear algebra), lab sciences (chemistry and physics), and engineering sciences (e.g., statics, thermodynamics). Where once there was only one FE component in the afternoon for all engineering disciplines, there are now several additional discipline-specific options, including ones for chemical, civil, electrical, industrial, mechanical, and petroleum engineering.

---

8. http://www.ieaust.org.au

After passing the FE exam, a person can be called an "engineer-in-training." Then, after four years of experience, the person must pass an NCEES Principles and Practice of Engineering (P&P) exam in his or her specific discipline. (The four years of experience requirement applies to graduates of ABET programs. Graduates of related programs usually need more experience.) At that time, a person can then apply for a PE license. Currently, there are no FE afternoon or P&P exams for software engineering.

Texas is the only state that currently licenses software engineers as PEs. The state has been doing so since September 1998 (Bagert, 1998b). As there are no software engineering licensing exams, the Texas Board of Professional Engineers has been using its waiver clause (which is available for all engineering disciplines licensed by the board) to license software engineers. In this case, practitioners with 12 years of experience and an ABET-accredited degree, or 16 years of experience for those with a related degree, are eligible to be licensed. All applicants in Texas, regardless of waiver status, must pass an engineering ethics examination.

In response to the Texas Board's actions, Barbara Simons, president of the Association for Computing Machinery (ACM), in March 1999, appointed an advisory panel to make recommendations to the ACM regarding the licensing of software engineers. On 15 May 1999, the ACM passed the following motion based on a report from the panel (Allen et al., 1999):

> ACM is opposed to the licensing of software engineers at this time because ACM believes that it is premature and would not be effective at addressing the problems of software quality and reliability.

As part of the fallout from this decision, the ACM Council on 30 June 2000 passed a motion to withdraw from SWECC, because (in the council's opinion) "SWECC has become so closely identified with licensing of software engineers under a professional engineer model" (Bagert, 2000).

Canada

In September 1999, Professional Engineers Ontario (PEO), the regulatory body for engineering in Ontario, announced licensing guidelines for software engineers, becoming the first (and to date, the only) province in Canada to do so (PEO, 1999). Up to this point, software practitioners had been assessed by PEO on an individual basis to see if they qualified for a Professional Engineer (P.Eng.) license.

PEO's requirements for a P.Eng. license for software practitioners include:

- graduation from an engineering program approved by CEAB, or equivalent education;
- four years of suitable employment experience (Applicants without an appropriate degree may demonstrate more extensive work experience in place of education, or may be required to write examinations.);
- knowledge of control theory, mathematical foundations, digital systems and computer architecture, software design and programming fundamentals;
- knowledge of three of the following—communications, optimization, data management, real time and control systems, performance analysis, parallel/distributed systems, and human interfaces and ergonomics;
- successful completion of the Professional Practice Examination on engineering law and ethics.

CERTIFICATION

ICCP

The Institute for Certification of Computing Professionals (ICCP)[9] is located in Des Plaines, Illinois, USA, and is under the direction of seven constituent societies, including ACM. ICCP offers a specialty exam in software engineering, which includes the following topics:

- computer system engineering,
- software project planning,
- software requirements,
- software design,
- programming languages and coding,
- software quality assurance,
- software testing techniques,
- software maintenance and configuration management.

IEEE-CS

In June 1999, the Computer Society began to work on a Competency Recognition Program. IEEE-CS plans to offer a number of certification exams

9. http://www.iccp.org

in a variety of areas and specializations (Tripp, 1999). The first of these exams, targeted to software engineering professionals with six or more years of experience, will be offered sometime in the first half of 2001.

Microsoft and Novell

Despite the fact that Microsoft is a software company, the Microsoft Certified System Engineer (MCSE) series of examinations actually concentrate on the installation, configuration, and troubleshooting of network systems, as does Novell's Certified Novell Engineer program.

SUMMARY AND FUTURE DIRECTIONS

During recent years we have seen dramatic changes in software engineering education, accreditation, licensing, and certification. Undergraduate software engineering degree programs are now being offered. ABET, as a result of the CSAB/ABET merger, is planning to accredit undergraduate software engineering programs. Some countries, states and provinces are licensing software engineers. Certification, which has existed in some form for years, is taking on increased importance as an alternative to licensing. All of these changes, and many associated activities, have taken place in the context of making software engineering a profession.

In the future we expect that there will be continued debate and continued evolution of software engineering as a profession. The exact role that licensing and certification will play is undetermined. The role of accreditation seems clear and is likely to be the same for software engineering as it is for other engineering fields. Accreditation of software engineering degree programs is one of the more mature processes of those under discussion in this paper.

ACKNOWLEDGEMENTS

Much of the background material on accreditation, licensing, and certification, as well as some of the material in the accreditation section, was drawn from a paper co-authored by A. Joe Turner (Mead, 1998). We would like to recognize Joe's contribution to this paper and to the profession.

REFERENCES

Allen, F., Boehm, B., Brooks, F., Browne, J., Farber, D., Graham, S., Gray, J., Hawthorn, P., Kennedy, K., Leveson, N., Nagel, D., Neumann, P., Parnas, D., & Wulf, B. (1999). ACM panel on professional licensing in software engineering report to council (http://www.acm.org/serving/se_policy/report.html).

Association of Universities and Colleges of Canada and Council of Professional Engineers. (1999). ‘‘Software engineering lawsuit discontinued.’’ Communique reprinted under the heading ‘‘Canadian lawsuit discontinued.’’ *FASE, 9*, 10 (http://www.cs.ttu.edu/fase/v9n10.txt).

Bagert, D. (1998a). Texas Board votes to license software engineers. *ACM Software Engineering Notes, 23*(5), 7.

Bagert, D: (1998b). Software Engineering Coordinating Committee formed. *FASE, 8*, 12 (http://www.cs.ttu.edu/fase/v8n12.txt).

Bagert, D. (2000). ACM withdraws from SWEcc. *FASE, 10*, 7 (http://www.cs.ttu.edu/fase/v10n07.txt).

British Computer Society and The Institution of Electrical Engineers. (1989). A Report on Undergraduate Curricula for Software Engineering. London, UK: Institution of Electrical Engineers.

Dupuis, R., & Bourque, P. (1999). Software engineering body of knowledge. *FASE 9*, 3 (http://www.cs.ttu.edu/fase/v9n03.txt).

Edwards, H. (1999). Software Engineering Education from a UK Academic’s Perspective. *Proceedings of COMPSAC 99*, Phoenix, AZ, 27–29 October, 243–244.

Ford, G., & Gibbs, N. (1996). *A mature profession of software engineering* (Tech. Rep. CMU/SEI-96-TR-004). Pittsburgh, PA: Carnegie Mellon University, Software Engineering Institute.

Gabrini, P. (1999). Canadian Computer Science Accreditation Council accreditation criteria for computer science and software engineering. *FASE, 9*, 3 (http://www.cs.ttu.edu/fase/v9n03.txt).

Gillespie, G. (1997). ABET asks the IEEE to look at software engineering accreditation. *IEEE—the Institute, 21*(7), 1.

Gotterbarn, D., Miller, K., & Rogerson, S. (1999). Computer Society and ACM approve software engineering code of ethics. *Computer, 32*(10), 84–88.

Grant, D. (2000). Undergraduate software engineering degrees in australia. *Proceedings of the 13th Conference on Software Engineering Education and Training*, Austin TX, 6–8 March, 308–309.

The Institution of Electrical Engineers. (2000). *Bulletin of Accredited Degree Courses, 23*, London, UK.

Mead, N. R. (1997). Issues in licensing and certification of software engineers. In *Proceedings of the 10th Conference on Software Engineering Education and Training* (pp. 150–160). Los Alamitos, CA: IEEE Computer Society Press.

Mead, N. R. and Turner, A. J. (1998). Current accreditation, certification, and licensure activities related to software engineering. In *Annals of software engineering*, Vol. 6, 1998, published April 1999 (pp. 167–180). The Netherlands: Baltzer Science Publishers.

Peters, D. (1999). Update on lawsuit about use of the term ‘software engineering’. *FASE, 9*, 3 (http://www.cs.ttu.edu/fase/v9n03.txt).

Professional Engineers Ontario. (1999). Engineering Association embraces software practitioners. Toronto, Ontario: Media Release (http://www.peo.on.ca/communications/Software_release.htm).

Shaw, M. (1996). An initial taxonomy for certification. Working paper. Pittsburgh, PA: Carnegie Mellon University.

Tripp, L. (1999). IEEE Computer Society competency recognition program. *FASE, 9*, 7 (http://www.cs.ttu.edu/fase/v9n07.txt).