Developing Software Forensics Standards for Collaborative eLearning Systems[♦]

Dr. Jinan A.W. FIAIDHI[*], Dr. Sabah M.A. MOHAMMED[*]
and
Dr. Kanaan A. FAISAL[#]

[*] Department of Computer Science, Lakehead University,
955 Oliver Road, Thunder Bay, Ontario P7B 5E1,
CANADA
jinan.fiaidhi@lakeheadu.ca
sabah.mohammed@lakeheadu.ca

[#] Department of Information and Computer Science, King
Fahd University for Petroleum and Minerals, P.O. Box 37,
Dhahran 31261, Saudi Arabia.
faisal@ccse.kfupm.edu.sa
manuel@ccse.kfupm.edu.sa

Abstract: A collaborative eLearning system is an extremely complex one and planning an effective e-infrastructure requires sound measures to ensure its integrity and security. This article attempts to address the issue of developing software forensics standards for intellectual property protection/identification based on watermarking within a university teaching environment.

Keywords: Software Forensics, Watermarking, Collaborative eLearning Systems

Collaborative eLearning Systems Trends:

Researchers have already identified the positive effects of social interaction during learning [1,2]. Furthermore, collaboration with other students has been shown to stimulate activity, make learning more realistic and to stimulate motivation. [3]. Research has also shown that moral dilemmas in computer ethics encourage group discussion and that teamwork encourages social facilitation, better learning and higher cognitive skills [4,5] and that groups can produce better solutions to moral and ethical problems than individuals can [6].

In most of eLearning programs offered today, the burden for learning is placed wholly on the shoulders of the learner. When a learner goes to a course web site, enters a grid that does not vary from course to course, consisting of menu of activities: announcements, documents, assignments, external links, communications, and tools. The course is served up as content that is devoid of any context. Everybody is expected to navigate this material on their own, without much support. The only extra help offered is email links to faculty and other students, but not much more. However, the marketplace is shifting in maturity. Currently these days everybody is expecting the arrival of the "second wave" of eLearning systems[7]. Such new systems should be first and foremost about creating a

[♦] This research is supported partially by the Canadian Bureau of International Education CBIE, Innovation Grant 2002/2003 and the Lakehead University NSERC Grant.

social space that must be managed for the teaching and learning needs of the particular group of people inhabiting that space. It should be collaborative in nature. Within education, collaboration is critical. Online, many students lack the contact of a face-to-face classroom. Activities that require collaborative work can put students in touch with each other, eliminating the sense of isolation that is common for first wave elearners. At it's simplest level, collaboration may be simply sharing information with another person, department, or organization...at it's most advanced level, collaboration involves unifying communication processes and content and establishing forums for accessing resources and building content and value together.

Collaborative eLearning systems merely rely on the internet for accessing learning materials [illegible] with [illegible]. Web-based collaborative learning means that the knowledge is not something that is delivered to learners, but rather something that emerges from active dialogue among those who seek to understand and apply concepts and techniques. Web-based collaborative learning systems can be divided into two categories, one is *asynchronous* system, and another is *synchronous*, which many practical systems were developed. The influential asynchronous system includes First Class, CSILE/Knowledge Forum, Learning Space, WebBoard, and WebCT; synchronous system includes Conference MOOS, WebChat Broadcasting System, and Microsoft Netmeeting. There are quite few notable technological tools which can be used to construct variety of web-based collaborative eLearning systems. Such tools includes Web Conferencing tools(e.g Centra, WebEx, PlaceWare, Latitude), Learning Management Systems (e.g Blackboard), Shared Display (e.g. VNC), Management Tools (e.g. TMD, NPAC), Learning Objects (e.g. IMS, ADL), Authoring Tools (e.g. Micromedia, VRML,SVG), Learning Portals (e.g. GEM, OGSA). There are many research in designing collaborative eLearning systems based on these tools[8,9].

The second wave should signals the arrival of greater standardization and the emergence of replicable processes besides enforcing collaboration. However, standardization of eLearning systems has many issues and central to all is the security infrastructure of such systems. elearning systems utilize the internet which is an open media of communication and collaboration presents difficulties with respect to security. The lack of the eLearning security and absence of intellectual properties of [illegible] other people work without proper acknowledgment to the source. The cases of student assignments plagiarism are increasing within a university teaching/eLearning environments[10,11]. Alternatively, software forensics copes with these problems by trying to identify the tool used to generate a particular intellectual property. In this paper we are developing Software Forensics standards for the collaborative eLearning systems environment.

Searching for Suitable Software Forensics Technique:

Software forensics is the use of authorship analysis techniques to analyse computer software for legal or official purposes[12]. Authorship analysis in literature has been widely debated for many years, and a large body of knowledge has been developed[13]. Authorship analysis on computer software, however, is different and more difficult compared to the other forensics paradigms. Many approaches attempted to analyze authorship based on statistical reasoning along with measures extracted from the writing style of the computer software (e.g. program source, email, text assignment)[14,15,16]. Pure stylistic approach cannot easily prove the identity of the author even if a positive correlation is found based on large sample of data[17]. With email software the case is somehow better because the email text body is not the only source of authorship attribution. Other evidence in the form of e-mail headers, e-mail trace route, e-mail attachments, the time stamps, etc. can, and should, be used in conjunction with the analysis of the e-mail text body[18]. Moreover, image authorship analysis has long researched and large success has been reported via image wavelets matching[40]. Generally in any eLearning system textual artefacts (e.g. programs, assignments) are most popular objects for collaboration.

Two more solutions has been used for authorship identification based on Cryptography and Watermarking. Cryptography is an old technology which can only protect the distribution of content and once a customer decrypts it, all protection is Lost. Such security technologies do not provide persistent security and are open to loss of income and intellectual property poaching[21]. Watermarking, on the other hand, is a relatively new technology which can compliment cryptography, providing protection after decryption, even when the content has entered the analog world. Watermarking involves embedding data, often imperceptibly, into a data medium or multimedia object to enhance or protect its value. While the watermarking field is relatively new, many applications that could benefit from watermarking have been proposed. The recent work of Mintzer et al [22,23] at the IBM Thomas Watsom Research Centre identified three clusters of applications with similar technical requirements: One uses watermarking to convey ownership information; another uses watermarking to verify that the object content has not changed; and the third, called collaborative watermarking, conveys object-specific information to a community of recipients. According to Mintzer[24] ACM Invited Paper, the three main emerging classes of applications lack standard marks, standard ways of interacting with systems, benchmarks tests, and even a standard terminology, thus presenting opportunities for developing application specific watermarking techniques.

Proposing a Watermarking Standards for Collaborative eLearning Systems:

Though digital watermarking of various types has been around since at least the early 1990's, the remaining open questions seem to be in a continuous-growth mode. Different Watermarks are not alike. Different techniques are used to embed different types of watermarks into digital media objects to accomplish different goals. In this article, we are proposing a methodology for standardising collaborative watermarking for the collaborative eLearning environment. Such methodology will enable the eLearning system to employ multiple watermarks to convey multiple sets of information, intended to satisfy differing or similar goals.

The first step of proposed methodology is to addresses the issue of functional requirements of Collaborative Watermarking. Generally, a watermark must convey as much information as possible. This implies that the watermark data rate should be high. A watermark should be a secret and be accessible to authorized parties only. This is can be achieved by the use of cryptographic keys. A watermark is an integral part of the data. It must persist even after signal processing and data manipulation. This also includes malicious manipulation that attempts to remove the watermark. This requirement is known as 'robustness requirement'. A watermark though being irremovable should also be imperceptible. It should not modify or alter the quality of the content. Normally, the degradation in quality is well below one percent. Watermark recovery process may not be allowed to use the original contents of the digital watermark. Most specifically for a collaborative environment, we need to embed several watermarks into the same digital object. Among such watermarks are ownership watermark, content integrity watermark and object-description (caption) watermark. The order of embedding these watermarks should not affect the robustness and fragileness of these watermarks. One safe order[24] is to embed the most robust ownership watermark first followed by moderately robust [illegible] and to embed the most fragile content verification last. The issue of effective watermarking algorithms can be determined from performing an intensive survey of the most recent algorithms and technologies [25,26].

In the *second step*, the methodology proposes to construct three generic watermarking utilities (e.g Browser plug-in [27]) which can manipulates watermarks from within popular document producing applications (e.g. MS Word, Notepad, Adobe, Corel, JBuilder). The *Watermark Embedders* - This software automatically adds the watermark to the digital object (May use Wavelets with appropriate parameters [28]), the *Watermark Readers* – To read and identify an embedded water mark, and the *Watermark Tracking* - This is software assists authorities in searching for watermarks in a particular server domain or in cyberspace. It should utilize the spider technology to search the Web for your water marked objects and report the findings back to you, so that you may take action against any inappropriate usage of your data. Such software is quite important for detecting plagiarism, which is a major concern in designing any collaborative teaching environment [10]. The next section shed the light on designing such Watermarking Tracking System.

In the *third step* a protocol for collaborative watermarking is proposed. Since every eLearning system utilises the TCP/IP protocol on the internet/intranet/extranet network, then we can think of modifying that protocol to convey watermarks. Handel & Sandford[29] and Wolf[30] found that the reserved or unused fields in the packet header can be used for information hiding. It is the basic layered design principle of the TCP/IP (OSI) network where the IP datagram encapsulate information received from the transport layer. In particular the IP header encapsulates ICMP messages and IGMP report and query messages. Covert channels in the Ipv4 header can, therefore also, be associated with those in the TCP, ICMP or IGMP headers. The Ipv4 header contains fragmentation information especially in the flag field (First bit is reserved, second bit DF (do not fragment, and the third bit MF(more fragments)). So in an unfragmented datagram, we can have 13 bit to hide information such as a watermark. Such redundancy provide us with a new venue to develop ICP/IP watermarks and to develop packet filters which can be used by the routers to reinforce its filtering policy.

In the *fourth step* a collaborative interfacing supervisor is constructed. An object content owner approaches a neutral registration authority of the eLearning system. Depending on the nature of object content, the authority allots a unique registration number. It also archives content and the unique registration number for future reference. A content owner generates a suitable watermark using primitives generated in step 2 and using a watermarking algorithm to embed it within the data. Such a watermark should be unobtrusive and secure. To ensure security of embedded digital watermarks, one or several secret and crypto logically secure keys can be used. To ensure robustness against data manipulation and processing, it is helpful to have very small digital watermarks and ensure that they are redundantly distributed in the host data. The digital watermark, public/private key and host data is processed using a watermarking algorithm to generate the watermarked data. To extract (detect) the watermark, the authorized agency requires watermark readers and a secure/public key. All these inputs are processed by the watermark recovery program to extract the watermark or confidence measure. The confidence measure indicates the degree of closeness of the original watermark and recovered watermark. The driving interface can deploy spread-spectrum communication[31] using the redundancy bits of the Ipv4 protocol. In such a scheme a watermark is embedded by adding pseudonoise (PN) signal. The collaborative supervisor also should manage a collaborative bus, which enables the eLearning system to deal with real-time event exchange, dynamic joining and leaving, concurrency control and crash recovery. The collaboration bus should include set of communication ports where peers can subscribe and publish information. Figure 1 illustrates the final collaborative architecture of the proposed methodology.

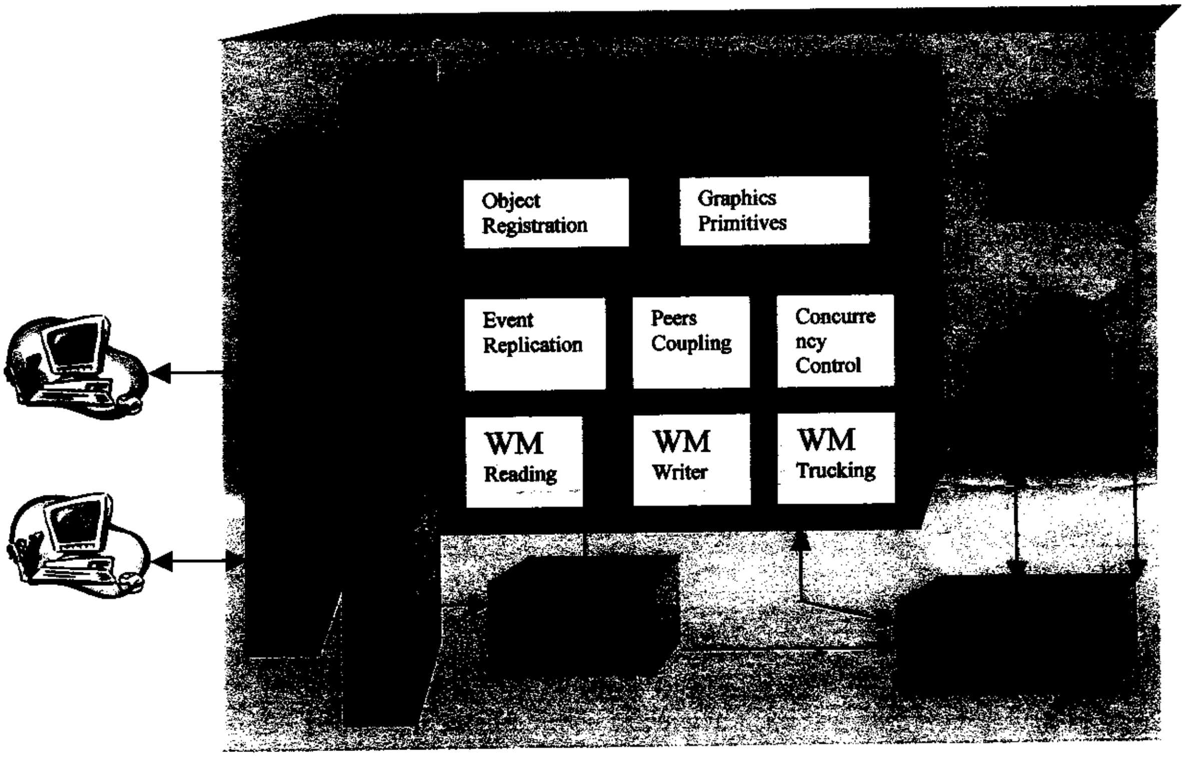

Figure 1: Proposed Secure Collaborative eLearning Architecture.

The collaboration supervisor design involves other modules that cooperate in maintaining a workable collaborative environment. The Advice and *Differences Resolver* detects opportunities for students to collaborate by finding significant differences between individual and group participation. The Differences Resolver starts by detecting differences specifically related to the currently added object, or find all "extra work" that the student can contribute to the group. The *Participation Monitor* attends to the activity in the group work. If nobody has worked in the group diagram for a period of time, it reports this event to the Adice and Differences Resolver to generate a proper advice. It also monitors whether each student is participating too much or too little. The Advice and Differences Resolver and Participation Monitor communicate their results to the *Collaboration Supervisor* via a *Blackboard.* The blackboard replaces a conventional whiteboard in a web-based educational environment. It provides learners and instructors with a board where they can introduce multimedia information, from text and images to video. Whatever data is introduced by any of the participants would be synchronously presented to the others. In order to identify the source for any data introduced in the board, every user is assigned a different color when he/she enters the current whiteboard session. Finally, the Collaboration Supervisor maintains an internal model of the environment and other knowledge via a database.

Software Forensics Watermarking Tracking Engine:
The watermarking tracking engine will enable us to detect cyberplagiarism cases. It will work as a personal assistant to the facilitators. This system represents a learning portal which has the ability to learn from its previous searches. Since the major software object that a collaborative eLearning system utilize is computer programs and assignments, then this tracking engine will deal with text search only. However, tracking images and images watermarks has been researched within the paradigm of Context-Based Image Retrival (CBIR) and there are sound image searching engine such as AMORE, QBIC, IMEDIA, VIPER, Virage, and Zomax [32].

Text search, on the other hand, relies either on dedicated text-based search engines (e.g dogpile.com, invisibleweb.com, reputes.com), or on some general-purpose plagiarism detection packages. The list of such general-purpose plagiarism detection packages include: Turnitin (www.turnitin.com), FindSame (www.findsame.com), Eve2 (www.CanNexus.com), CopyCatch (www.CopyCatch.freserve.co.uk) and WordCHECK (www.wordCHECK.com). The UK Joint Information System Committee (JISC) surveyed these packages from user and technical perspectives and concluded that they are very limited in detecting material that is cut and pasted from the Internet[19] and there is no dedicated package that can help in detecting plagiarism of programming source code within cyber space. What is actually available are a few packages that help can in comparing a given set of programs and report suspected cases of similarity. Most notable among those packages are: MOSS, YAP3, and JPlag. All such packages presented very weak measures of similarity for authorship identification and cannot work for a variety of programming languages as well as within a collaborative learning environment[20].

Hence it is believed that text searching and text-based watermark trucking depends upon development and use of relevant metadata standards. Text-based Metadata is usually defined as data about text. In order to allow more intelligent syndication, we propose that our search engine should utilizes an ontology to narrow their searching space. This ontology can provide an expressive terminology for describing content, and inferences sanctioned by the ontology and can be used to improve the quality of search on the tracking engine[33]. The initial ontology can be extracted from the problem description or from a typical solution. Figure 2 illustrates an overview of its essential components.

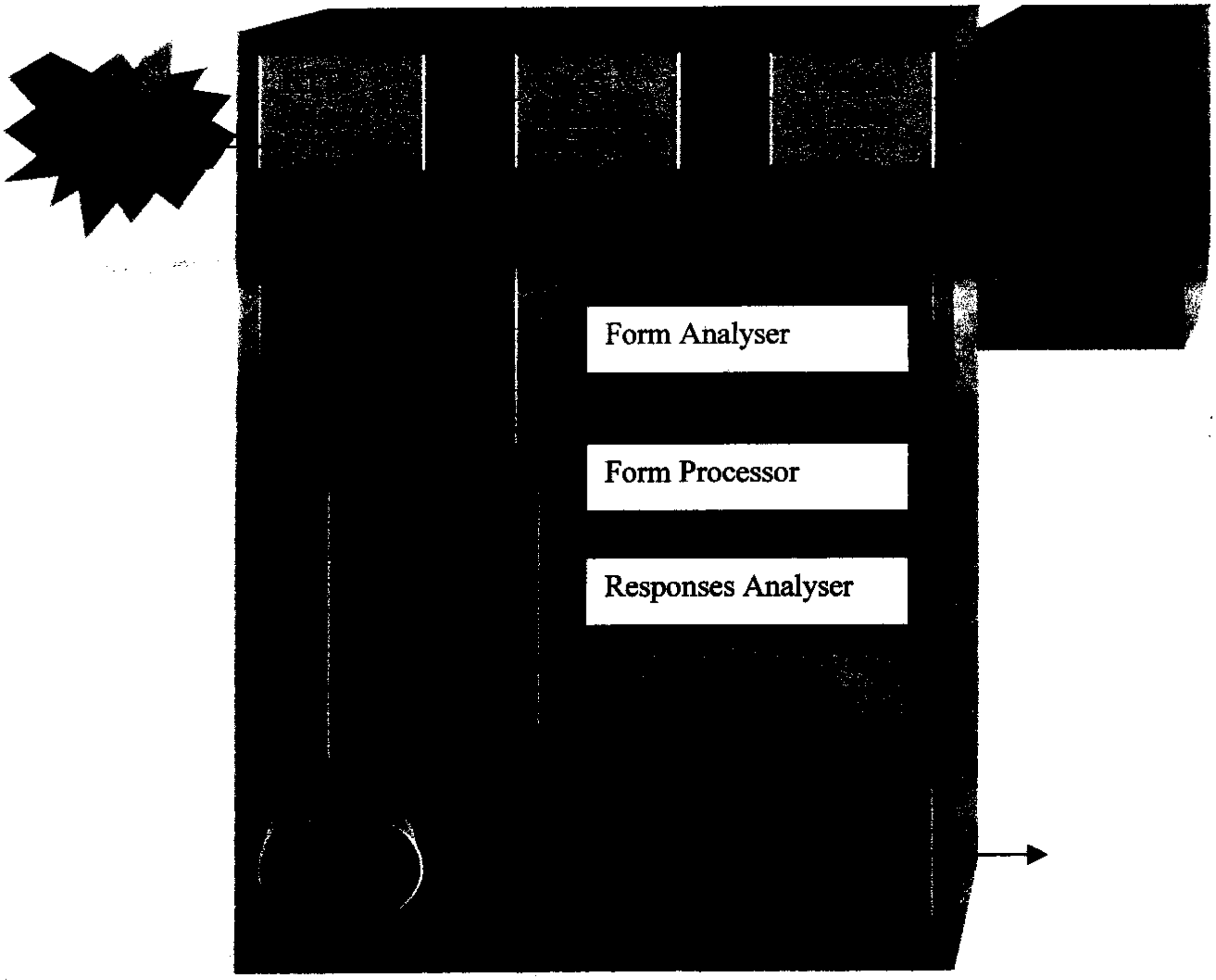

Figure 2: An Overview of the Proposed Watermarking Tracking Engine.

The tracking engine make use of various other computer programs that trace hyperlinks across the Web. Besides the typical HTTP server (e.g.TomCat) the query handler module, which formulates queries from the extracted ontology. Also it has a deep crawler module which uses the queries to follow links across the deep,web[34]. The crawler may start with initial set of URLs according to the instructor's experience. The crawler tries to find relevant text pieces within databases accessed only by their web site interface. The form analyser will try to match from the available ontology's certain matching substitutions and then pass it to the form processor to send it and its responses is analysed at the responses analysis module. The extracted pages are then passed to the filtering intelligent agent. The crawler also passes the retrieved pages to a local repository. The crawler continues visiting the web, until the importance of their links becomes below certain threshold[35]. The filtering intelligent agent determines what to visit next, and feeds the links back to the crawler. The filtering agent passes those relevant pages to both the indexer and collection analysis modules. The indexer module extracts all the words from each retrieved page and records the URL where each word occurred. The indexer stores the URLs in special lookup storage and directs the words into a special lexicon. The collection analysis module, however, separates the retrieved pages into classes according to the required field of search. Finally, the plagiarism determiner searches for known watermarks within all the retrieved pages from a specialized databases with all the existing students assignments/programming code available at a local store. For those retrieved programs that posses no watermarks one can use other measures such as: students logs, reflection-based measures[36], the lexical chains[37], UML flow diagrams[38], and the exceptions flow[39]. The full set of measures and their affectivity is still under investigation. The suspected cases of plagiarism whether occurred locally or from the cyberspace are reported back to the collaboration supervisor for proper facilitator advice.

Conclusions

Despite the ongoing development and growing sophistications of collaborative eLearning system, the issue of security standards has not been addresses properly. There are large number of collaborative eLearning system where security is not a design issue, including CSILE,WebBoard, WebCT, WebChat, Centra, WebEx, Blakboard, TDM, NPAC, IMS, Micromedia, GEM. There are two ways to achieve security: cryptography and watermarking. The computer industry has not yet agreed upon a universal standard for digital watermarks. This article address the issue of establishing standards based on watermarking for constructing secure collaborative eLearning systems. The article provides a methodology or framework for enforcing forensics standards: embedding, detecting and searching for digital watermarks. The framework starts from the design level and ends with a proper system architecture. This article also provides another framework for tracking text-based watermarks through the use of the ontology extracted from the learner logs and the facilitator problem description. The tracking framework utilizes a deep crawler for searching cyber-based databases. Such frameworks and standards are currently used to establish a collaborative eLearning environment within the new established Advanced Technology Academic Centre (ATAC) of Lakehead University. The authors aim to use these standards to construct a Java Beans-Based Collaborative system (JBBC). The implementation of such system will use programming concepts developed by Marsic[41].

References

1. Crook, C. *Computers in the community of classrooms.* In K. Littleton, & P. Light (Eds.) Learning with computers. Analysing productive interaction. London and New York: Routledge, 102-117. (1999)
2. Dillenbourg, P. *Introduction: What do you mean by "collaborative learning"?* In P. Dillenbourg (Ed.) Collaborative learning. Cognitive and computational approaches. Advances in Learning and Instruction Series, Amsterdam: Pergamon, 1-19. (1999)

3. Veerman, A. & Veldhuis-Diermanse, E. *Collaborative learning through computer-mediated communication in academic education.* Paper presented at Euro CSCL conference, Maastricht, Holland. (2001)
4. Hiltz, S.R. *The Virtual Classroom: Learning without limits via computer networks.* Ablex Publishing. Norwood, New York. (1994)
5. Saloman, G. and Globerson, T. *When teams do not function they way they ought to.* Journal of Educational Research, 13(1), 89-100. (1989)
6. Peek, L.E., Peek, G.S. and Horas, M. *Enhancing Arthur Andersen Business ethics Vignettes: Group Discussions using Cooperative/Collaborative Leaning Techniques.* Journal of Business Ethics, 13, 189-196 (1994)
7. Taylor, C., eLearning: The Second Wave, Learning Circuits OnLine Journal, October 2002 (www.learningcircuits.org).
8. Jianhua Z., Kedong L., Akahori A., Modelling and System Design for Web-Based Collaborative Learning, Int. Conference on Information Technology Baed Higher Education and Training, July 4-6, 2001, Kumamoto, Japan.
9. Anido L, Liamas M, Fernandez M, and Caeiro M., An Environment for Web-Based Collaborative Life Long Learning, 16th IFIP Word Congress In. Conference, China, ICEUT2000, China July 2000.
10. Fiaidhi,J and Robinson,S., Similarity analysis and Plagiarism detection in a university teaching environment, Int. J. Computer and Education, Vol. 11, No. 1,1987.
11. Fiaidhi J, Mohammed S, and ALKhanjari Z, Designing an On-Campus Learning Portal, Journal of Science and Technology, Vol 6, No.1, 2001.
12. Spafford and Weeber, Software Forensics: Can we track code to its author?, Purdue University technical Report # CSD-TR 92-010, Feb 1992.
13. Dauber. K *The Idea of Authorship in America.*The University of Wisconsin Press, 1990.
14. Vel, O et al , Multi-Topic email Authership Attribution Forensics, ACM Conference on Computer Security- Workshop on Data Mining for Security Applications, Nov 2001, Philadelphia, PA, USA.
15. Krsul I and Spafford, E, Authorship Analysis: Identifying the Author of a Program, Proc. 18th {NIST}-{NCSC} National Information Systems Security Conference", pp514 524,1995.
16. Oman P and Cook, C., Typographic style is more than cosmetic. *Communications of the ACM*, 33(5):506–520, 1990.
17. Wong, J Kirovski, D and Potkonjak, M Computational Forensics Techniques for Intellectual Property Protection, Information Hiding Workshop, Pittsburgh, PA, April 2001.
18. Vel, O, Mining email Authership, KDD-2000 Workshop on Text Mining, August 2000, Boston, USA.
19. Bull, J. et al., Technical Review of Plagiarism Detection Software Report, JISC Report, University of Luton, 2001 (available at www.jisc.ac.uk/pub01/luton.pdf).
20. Culwin, F, MacLeord, A, Lancaster, T., Source Code Plagiarism in UK HE Schools, JISC Technical Report, SBU-CISM-01-01, South Bank University, September, 2001 (available at www.jisc.ac.uk/pub01/southbank.pdf).
21. Wayner P, Disappearing Cryptography, 2nd Edition, Morgan Kaufman, 2002

22. Mintzer, F, Safeguarding digital library, DLIB Mag. Dec.1997 (www.dlib.org/dlib/december97/ibm/12lotspiech.html)
23. Mintzer, F, Opportunities for watermarking Standards, CACM, Vol 41, No. 7, July 1998
24. Mintzer, F. et al., Effective and Ineffective Image Watermarks, IEEE 1997 In. Conf. On Image Processing, Vol. III, paces 9-12, October 1997.
25. Yeung, M., Digital Watermarking, CACM, Vol 41, No. 7, pp31-33, July, 1998.
26. Cox,I and Miller, M, The first 50 years of watermarking, J.Applied Signal Processing, Vol. 2, 2002.
27. Jenkin M and Dymond P.,A Plugin-Based Privacy Scheme for WWW file [illegible]
28. Dietze M and Jassim S.,The Choice of Filter Banks for Wavelet-Based Robust Watermarking, ACM Multimedia Workshop,NY,2002.
29. Handel M and Stanford D, Hiding data in the OSI network, 1st Int. Workshop on Information Hiding, Cambridge,UK, May-June 96.
30. Wolf, G, Covert channels in LAN protocols, Proceedings of the Workshop on Local Area Network Security (LANSEC'89),pp91-102,1989.
31. Viterbi, A, CDMA: Principles of Spread-Spectrum Communications, Addison-Wesley,1995.
32. Mirmehdi M. and Perissamy R., Perceptual Image Indexing and Retrieval. *Journal of Visual Communication and Image Representation*, 13(4):460--475, December 2002.
33. Luke S., Spector L, Rager D, Hendler J, Ontology-Based Web Agents, Proceedings of the 1st Int. Conference on Autonomous Agents (Agents'97), Marina del Rey, CA, USA
34. Menczer F. et al, Evaluating Topic-Driven Web Crawlers, ACM SIGIR'01, Sept 9-12, 2001, New-Orleans,USA.
35. Cho, J, Garcia-Molina H, and Page,L, Efficient Crawling through URL ordering, Proceedinggs of the 7th Conference WWW98, 1998.
36. Welch I., Stroud R., Kava - Using Bytecode Rewriting to add Behavioural Reflection to Java, Proceedings of USENIX Conference on Object-Oriented Technology UK (2001).
37. Hirst G.and Budanitsky A., Lexical Chains and Semantic Distance, Eurolan-2001, August 2001, Iasi, Romania.
38. Niere J., Wadsack J., Zündorf A.: Recovering UML Diagrams from Java Code using Patterns. accepted for Proc. of 2nd Workshop on Soft Computing Applied to Software Engineering, Enschede, The Netherlands, Lecture Notes in Computer Science, Springer, 2001.
39. Sinha,S and Harrold,M., Criteria for testing Exception-Handling constructs in Java Programs, IEEE Proceedings of the International Conference on Software Maintenance, Oxford-England, August-September, 1999.
40. Antoni M., Barlaud M., Mathieu P., and Daubechies I., Image Coding using Wavelet transform, IEEE Trans. On Image Processing, 1(2):205-220, April 1992.
41. Marsic, I., A Collaborative-Enabling Framework for Java Beans, Proc. Hawaiian In. Conference on Systems Sciences, HICSS-35, May, 1999.