國立雲林科技大學
96學年度博士班招生入學考試試題

系所別：資訊管理系
科　目：資訊管理文獻

Questions: 題目分成 A、B 二部份，作答時，請注意各題之比例配分並標示題號

Part A: (30%)

Please briefly describe what the major contributions, in MIS field, of the following authors are:

1. Orlikowski, W. (10%)
2. Delone, W.H. & McLean, E.R. (10%)
3. Davis, F.D. (10%)

Part B: (70%)

1. Please give the abstract, within 200 words in English, of this paper? (10%)

2. Draw the framework of this research, and then briefly describe major research variables. (10%)

3. Describe briefly the procedure of qualitative content analysis and supercode analysis conducted in this research. (10%)

4. Comment on "whether CMC moderates or magnifies the gender difference reported in face-to-face research", based on literature reviewed in this paper on both sides, as well as your own opinion. (10%)

5. Based on findings and insights from this paper suggest ways to implement an e-learning environment with more gender-free, equal-participation and supportive self-disclose characteristics. (10%)

6. Analyze potential mediating variables that might bring out different conclusions with those of this research. (10%)

7. Please explain why the author constructed 21 task codes, 12 linguistic codes, 16 stylistic codes, and 8 paralinguistic codes during his pilot study? (10%)

國立雲林科技大學
96學年度博士班招生入學考試試題
系所別：資訊管理系
科 目：資訊管理文獻

Available online at www.sciencedirect.com
ScienceDirect
Computers in Human Behavior
ELSEVIER
Computers in Human Behavior 23 (2007) 2240–2255
www.elsevier.com/locate/comphumbeh

Students' linguistic behaviour in online discussion groups: Does gender matter?

Jane Guiller *, Alan Durndell

Department of Psychology, Glasgow Caledonian University, 70 Cowcaddens Road, Glasgow G4 0BA, Scotland, United kingdom

Available online 11 May 2006

Abstract

1. Introduction

The increased use of networked computers in contemporary society has changed the ways in which students can engage in peer-based discussion. Consequently, higher education establishments are utilising virtual learning environments such as Blackboard and WebCT to support existing face-to-face undergraduate courses. A feature of such systems is their ability to support many-to-many communication in the form of text-based discussion forums. Asynchronous computer-mediated communication (CMC) refers to communication that takes place over computers in delayed time (e.g., email), in contrast to synchronous CMC (e.g., chat rooms), which takes place in real time. Hence, asynchronous CMC allows people to communicate from different places and at different times. A frequently cited educational advantage of the asynchronous nature of text-based online discussion forums is the time available to read a message and formulate a response, which can improve reflection upon and development of a topic (e.g., Harasim, 1990).

Early psychological studies of the Internet focused on what was lost in text-based CMC. It was predicted that the reduction in social context cues through visual anonymity would greatly reduce the capacity to transmit social and interpersonal information (e.g., Short, Christie, & Williams, 1976; Sproull & Kiesler, 1986). It was theorised that self-awareness was reduced in CMC. Thus, the 'cues-filtered out' approach (Sproull & Kiesler, 1986) predicted that communication will be depersonalised, less 'social' and more uninhibited, in comparison to face-to-face communication. It was assumed that the medium only had the capacity to transmit task-oriented and cognitive material and not socioemotional content. Socioemotional content includes the use of emoticons, expressions of supporting references, self-references, references to others and self-disclosure (Jaffe, Lee, Huang, & Oshagan, 1999).

In contrast to the 'cues-filtered-out' approaches, Walther's (1992) social information processing theory maintained that text-based CMC could support socioemotional and relational communication. This theory assumes that CMC users are affected by the same internal drive of 'affiliation', i.e., interaction with other people, as participants in other communicative contexts (Jaffe et al., 1999). Therefore, it is suggested that CMC users will adapt existing communicative cues, within restrictions of language use and textual display (e.g., paralanguage), in order to convey relational communication. However, a more recent theory of CMC known as the 'rational actor approach' (e.g., Markus, 1994), despite disputing the technological determinism of the 'cues-filtered-out' approach, is in keeping with it in terms of the proposed reduction in interpersonal communication. This has implications for the use of CMC in educational contexts as it suggests that students may be less responsive to the ideas of their peers during CMC, questioning the extent of 'affective learning' (see Biggs, 1987) that can occur in computer-mediated contexts.

Gender may also influence the socioemotional content of computer-mediated discourse. Males are believed to value status more through the process of gender role socialisation, whereas females are thought to value connection or affiliation more, leading to gender-preferential communication styles, differentiated as 'competitive' and 'cooperative', respectively (Coates, 1993; Tannen, 1991). Therefore, females may be more likely than males to express socioemotional responses in CMC, such as expressing support and disclosing opinions, feelings and experiences. However, negative forms that would also qualify as responses to another's ideas, such as expressions of disagreement, are not always included in definitions of socioemotional content.

The absence of social context cues in CMC, i.e., the lack of non-verbal indicators of hierarchy, status and power was also presumed to have a democratising effect on communication, leading to equalised participation (Sproull & Kiesler, 1986). It was suggested that when such cues are missing in CMC, participants who would otherwise defer speaking turns to higher-status participants in face-to-face interaction, become uninhibited and participate more. Therefore, studies began to emerge which examined participation in CMC by gender (e.g., Graddol & Swan, 1989; Selfe & Meyer, 1991), as it had been reported that males dominated mixed-sex interaction in formal face-to-face contexts by speaking more frequently, for longer and interrupting more, than females (e.g., Tannen, 1991; Thorne, Kramarae, & Henley, 1983).

It was suggested that the visual anonymity offered in text-based CMC would result in gender-free equality online and Graddol and Swan's (1989) findings supported this claim initially, as they found equal participation in an anonymous Open University conferencing system. However, this was later shown to be prematurely optimistic as Herring (1993, 1994) reported that males dominate online interaction by making longer and more frequent postings than females and subsequent studies, conducted in a variety of CMC contexts, supported this claim (e.g., Barrett & Lally, 1999; Richardson & French, 2000; Sierpe, 2000; Stewart, Shields, Monolescu, & Taylor, 1999).

Similar to face-to-face research, these results suggest that males dominate interaction, which may deter women from participating in CMC or force them to seek women-only groups online. Indeed, Yates (2001) suggested that altering the social context in the form of women-only online groups can moderate some of the problems experienced by women in educational interactions. However, gender segregation is neither practical, nor desirable, in an educational context. Furthermore, Miller and Durndell (2004) studied participation in an educational context and found that males and females were similar regarding quantitative measures of participation, as no significant gender differences were found in relation to the frequency or length of online postings. This was partly explained through the motivation to participate as marks were on offer for participation in this educational context and also because of the unequal gender balance, as females outnumbered males on the course. However, other studies have shown that relatively few males have still managed to dominate a discussion online (e.g., Herring, 1993; Light, Nesbitt, Light, & Burns, 2000; Sierpe, 2000).

For example, Light et al. (2000) studied purely student–student interaction via asynchronous CMC in the context of a third-year undergraduate course and found that in one group in particular the discussion became dominated by two males 'who shifted the discourse style from more formal contributions to 'testing the boundary' comments' (p. 94). These comments became increasingly directed towards particular individuals and the researchers report that there was evidence of inciting gender reaction 'for fun'. This shows that measuring surface variables in relation to the amount of participation does not address the extent to which participants can dominate through their use of language and interaction style online. Perhaps the presence of an instructor in these groups would have influenced the outcome of this study as Herring (1999) claims that female students participate more and sometimes more than male students in online classrooms in which the teacher controls the interaction, even when the teacher is male.

The removal of status cues such as gender has the potential to moderate the effects of gender on language use online, as Carli (1990) postulated that status may be an important determinant of gender differences in language use. This is known as the 'dominance' approach to gender and language (e.g., Poynton, 1985), and is often contrasted with the 'difference' approach, advocated by Tannen (1991). It is maintained that status is more of an influence on language use than gender. However, males are traditionally awarded more status and power in society and Sussman and Tyson (2000) suggested that these gendered power differentials in communication style would transfer into computer-mediated environments.

It was suggested that under conditions of anonymity males and females may be less likely to feel that they have to project the socially expected qualities corresponding to their gender (Matheson & Zanna, 1990). However, George (1995) claimed that communication is not neutral and texts are arenas in which to exercise power within social relationships. Moreover, Yates (2001) stated that 'CMC interaction, as with face-to-face communication, is founded upon existing social structures and perceptions which underlie inequalities in the interaction' (p. 27). For example, language may be a means of constructing and maintaining gendered power differentials in society. Therefore, CMC could potentially magnify, as opposed to moderate, the gender differences reported in face-to-face research.

The social identity explanation of deindividuated effects (SIDE) model (Postmes, Spears, & Lea, 2000, 2002; Reicher, Spears, & Postmes, 1995) challenged the anti-normative and deterministic view of the 'cues-filtered-out' approach described earlier. It predicts that visual anonymity of the self to others leads to heightened self-awareness and greater adherence to group norms when a social identity is salient. So, for example, if gender is made a salient social identity in CMC then this could invoke behavioural norms and stereotypes regarding gender appropriate behaviour, influencing expectations and perceptions of CMC users. Similarly, Matheson and Zanna (1990) argued that differences in status may actually be accentuated in CMC if cues to gender are available. Low public-awareness levels in CMC (i.e., decreased concern about others' impressions) are associated with lower social pressures that make the expression of internalised gender biases unacceptable. Therefore, it is possible that the exclusive focus on language in text-based CMC could not only exacerbate existing asymmetrical power differences, but even create them.

Matheson and Zanna (1990) found that CMC users reported greater levels of private self-awareness than participants communicating face-to-face. Heightened private self-awareness in CMC is associated with high levels of self-disclosure (Joinson, 2001). This may have implications for the use of visually anonymous and pseudonymous CMC in education as students may be more likely to disclose under these conditions. Furthermore, the sensitive nature of some psychological topics such as mental health, child abuse and eating disorders under these conditions may be conducive to self-disclosure. This may have implications for the use of CMC in psychology courses. As suggested earlier, it is also possible that the level of self-disclosure online may be influenced by gender, in line with traditional sex-role expectations. It could also be the case that in CMC self-disclosure is normative, in the same way that flaming (i.e., swearing, insults and hostile comments) is likely to be context-dependent and normative (see Lea, O'Shea, Fung, & Spears, 1992), as opposed to simply an outcome of the loss of face-to-face cues (e.g., Kiesler, Siegel, & McGuire, 1984).

Gender-related patterns in language style have been found in Internet discussion groups such as email listservs (e.g., Herring, 1993, 1994). Herring reported large differences in language style along the same task-orientated versus socioemotional dimensions as Tannen (1991) had described in face-to-face contexts. Female postings tended to display features of attenuation, such as hedging, apologising, asking questions and a personal orientation, revealing thoughts and feelings and interacting with and supporting others. On the other hand, male postings were lengthy and/or frequent, adversarial and featured strong assertions, self-promotion, sarcasm and flaming. Herring suggested that gender-based communication styles and the power dynamics associated with these styles carry over to electronic environments, despite the loss of overt face-to-face cues to gender.

Savicki, Lingenfelter, and Kelley (1996) provided some evidence for the gender-related communication styles identified by Herring (1994) in their study of Internet discussion groups. Soukup (1999) observed traditional masculine and feminine forms of discourse in Internet chatrooms. However, Yates (1997) notes that many studies have failed to replicate these findings. Moreover, Michaelson and Pohl (2001) did not find differences along the supportive/emotional versus adversarial/task-oriented dimensions in their study of an email problem-solving task.

Thomson, Murachver, and Green (2001) conducted experiments into the dynamism of linguistic exchange. They investigated how males and females accommodate to gender-preferential language in emails and found that participants used 'male' language when communicating with a male netpal and 'female' language when communicating with a female netpal, supporting the process of accommodation in online dyadic exchanges. In relation to many-to-many exchanges, where CMC participants can be exposed to a variety of styles, Herring (2003) suggests that the majority gender group online will have the greater influence on the shared discursive norms.

Mixed results in terms of gender variations in language use in CMC make it unclear whether CMC moderates or magnifies the gender differences reported in face-to-face research. Perhaps these differences could be attributed to the varying online contexts that have been studied. There is also a methodological issue in terms of the extent to which gender can reliably be inferred by researchers from usernames or email addresses. For example, Jaffe et al. (1999) found that females are more likely to choose a pseudonym to mask their gender when communicating online, which could distort the findings of studies using samples taken directly from the Internet. Miller and Durndell (2004) reported that females were significantly more likely to opt for a numerical pseudonym instead of their real name, in comparison to males.

Although text-based, pseudonymous CMC may be considered 'gender-free', cues to gender may not be restricted to usernames and signatures in CMC. Experimental work by Thomson and Murachver (2001) found evidence for gender-preferential language in informal CMC in a series of experimental studies, showing that readers of email messages used gender-linked language differences within messages to correctly identify the author's gender. Therefore, if linguistic cues to gender are found in real-life CMC contexts, this will have implications for the assumption of gender anonymity online and CMC as a gender-free environment.

The present research is concerned with formal use of CMC on an undergraduate psychology module. Ethical clearance permitted access to background details of the students, so that gender could be reliably inferred in this study. Education is a context in which gender equality is an important issue. If differences in communication style are found to exist in this context then this may have implications for the increasing use of CMC in Higher Education. Yates (2001) provided an overview of the research literature on CMC, gender, language and education and concluded that there is a lack of few large and detailed studies on this topic. This paper reports the results of an extensive project, employing both quantitative and qualitative techniques, that was carried out to investigate the existence of gender-related patterns in language use and interaction style in educational, mixed-gender, online discussion groups.

2. Method

2.1. Participants

The participants were 197 campus-based introductory psychology students (149 females, 48 males) at a Scottish university who had all chosen to take part in the online discussion groups as an optional part of their course. The age range was 17–46 years and the mean age of the sample was 22 years (SD = 6.52).

2.2. Coding scheme

The coding scheme was developed in Atlas.ti 4.2 (computer-assisted qualitative data analysis software) using a qualitative content analysis (QCA) procedure (Mayring, 2000). QCA seeks to conserve some methodological advantages of quantitative content analysis and broaden them to a concept of qualitative procedure. The unit of analysis was at the level of the whole message, resulting in a macro-analysis of whole messages and episodes of interaction, as opposed to a micro-analysis of single words and phrases, as is common in traditional quantitative content analysis.

During a pilot study, the author constructed a code list consisting of 21 task codes (e.g., initiates thread), 12 linguistic codes (e.g., intensifiers), 16 stylistic codes (e.g., challenging) and 8 paralinguistic codes (e.g., capitalisation), which included definitions and examples from the corpus. The majority of linguistic (see Appendix A) and stylistic codes (see Appendix B) were coded in previous studies as potential discriminators of gender (e.g., Herring, 1994; Savicki et al., 1996). The code list was then passed to a second independent rater who coded 20% of the sample (65 postings chosen at random) for 30 of the linguistic and stylistic variables. The second coder did not code for the more objective features such as 'first person pronouns' and 'intensifiers'. The inter-rater reliability measure for the more subjective codes such as 'humour' and 'rhetorical questions' was significant ($\kappa = 0.94$).

After this first level coding, second level coding involved the organisation of code families in Atlas. Codes were grouped into eight major code families. These were female language (e.g., self-disclosure, intensifiers), male language (e.g., humour, rhetorical question), task-orientated (e.g., answers question), socioemotional (e.g., references by name), attenuated (e.g., personal opinion, qualifiers), authoritative (e.g., strong assertion, presuppositions), negative socioemotional (e.g., disagreement) and positive socioemotional (e.g., agreement). A full description of these code families and examples is in Miller (2004). Analyses were then conducted on 'supercodes'. These are stored queries constructed from combinations of code families using Boolean operators in Atlas.ti 4.2 (e.g., 'return all messages coded as containing authoritative language but no attenuated language'). These higher-order codes permit analyses to be conducted into the use of combinations of codes and code families by males and females, allowing the identification of patterns of language use and particular interaction styles.

2.3. Procedure

Students on an introductory psychology module were invited to participate in online discussion groups for coursework credit. They were supplied with instructions on how to access the online forums and told that they must give either their real name or matriculation number as their user identification when posting messages in order to receive marks for contributing to the discussion. This allowed the gender of participants to be reliably inferred. Informed consent was gained from the online participants to download their contributions for analysis. Ethical clearance permitted access to background details of the students, such as gender. The online discussion forums covered a variety of introductory psychology topics such as eating disorders, intelligence, memory and personality. A total of 699 student postings were collated over four semesters. The mean number of posts was 3.55 and the average length of posts was 126.03 words. Tutors also contributed online but only the student contributions were analysed. Coding was carried out using Atlas.ti 4.2, as described above. Statistical analysis was conducted using the χ^2 Test and Fisher's Exact Test in SPSS 10. Yates' correction was applied to cells that have expected frequencies of less than five.

3. Results

A total of 699 postings (538 female, 161 male) were analysed. The electronic discourse was characterised by extensive first person pronoun usage as 87% of all postings analysed contained some form of first person or plural pronouns (e.g., 'I', 'we'). Also, 38% of student contributions directly responded to the ideas of other students by expressing agreement and/or disagreement. Just over a fifth of all postings contained references to own emotions (e.g., 'I am angry that') or self-disclosure (e.g., 'I myself have been overweight').

Table 1 above shows participants' use of the linguistic variables by gender. The table lists each linguistic variable and the corresponding proportions of males and females making each type of contribution, on which χ^2 analyses were conducted. The percentages of each type of posting, expressed as a percentage of the total male and female postings are also given in the table. As can be seen from the table, a significant result was obtained for only one individual linguistic variable, namely intensifiers. It was found that significantly more females used intensifiers in their postings than males, $\chi^2(1, N = 197) = 6.77, p < 0.01$.

Table 2 shows participants' use of stylistic variables by gender. As before, the stylistic variables are listed, as are the corresponding proportions of males and females making each type of contribution, on which χ^2 analyses were conducted. It can be seen from the table

Table 1
Participants' use of linguistic variables by gender: χ^2 of male and female samples

	% of participants		χ^2	% of postings	
	Male ($N = 48$)	Female ($N = 149$)		Male ($N = 161$)	Female ($N = 538$)
Absolute adverbials	29% ($N = 14$)	26% ($N = 39$)	0.16	12% ($N = 19$)	11% ($N = 57$)
Exhortations	4% ($N = 2$)	7% ($N = 11$)	0.11	1% ($N = 2$)	2% ($N = 12$)
First person plural pronouns	54% ($N = 26$)	55% ($N = 82$)	0.42	30% ($N = 49$)	28% ($N = 153$)
First person pronouns	83% ($N = 40$)	92% ($N = 137$)	2.46	70% ($N = 113$)	86% ($N = 461$)
Second person address	25% ($N = 12$)	26% ($N = 38$)	0.00	18% ($N = 29$)	12% ($N = 64$)
Imperative verbs	21% ($N = 10$)	15% ($N = 22$)	0.98	7% ($N = 12$)	6% ($N = 30$)
Impersonal truths	17% ($N = 8$)	8% ($N = 12$)	2.46	6% ($N = 10$)	2% ($N = 12$)
Intensifiers	56% ($N = 27$)	76% ($N = 113$)	6.77*	41% ($N = 66$)	50% ($N = 270$)
Interjections	10% ($N = 5$)	15% ($N = 22$)	0.58	4% ($N = 7$)	7% ($N = 36$)
Qualifer/hedge/tag question	46% ($N = 22$)	55% ($N = 82$)	1.23	27% ($N = 43$)	30% ($N = 163$)
References to emotion	35% ($N = 17$)	46% ($N = 69$)	1.75	15% ($N = 24$)	24% ($N = 127$)
Rhetorical questions	33% ($N = 16$)	32% ($N = 47$)	0.05	15% ($N = 24$)	13% ($N = 71$)

* Significant at $p < 0.01$.

Table 2
Participants' use of stylistic variables by gender: χ^2 of male and female samples

	% of participants		χ^2	% of postings	
	Male ($N = 48$)	Female ($N = 149$)		Male ($N = 161$)	Female ($N = 538$)
Agreement	21% ($N = 10$)	48% ($N = 72$)	11.30**	9% ($N = 15$)	26% ($N = 142$)
Challenging	38% ($N = 18$)	10% ($N = 15$)	19.59***	17% ($N = 27$)	4% ($N = 20$)
Compliments	10% ($N = 5$)	23% ($N = 34$)	3.52	4% ($N = 6$)	10% ($N = 52$)
Controversial	15% ($N = 7$)	1% ($N = 2$)	14.14***	5% ($N = 8$)	0.4% ($N = 2$)
Disagreement	44% ($N = 21$)	15% ($N = 22$)	17.86***	19% ($N = 30$)	6% ($N = 30$)
Empathic	2% ($N = 1$)	13% ($N = 20$)	4.91*	1% ($N = 2$)	4% ($N = 23$)
Humour	31% ($N = 15$)	7% ($N = 10$)	19.74***	12% ($N = 20$)	2% ($N = 13$)
Personal experience	23% ($N = 11$)	40% ($N = 59$)	4.41*	9% ($N = 14$)	19% ($N = 100$)
Personal orientation	60% ($N = 29$)	72% ($N = 108$)	2.49	39% ($N = 63$)	51% ($N = 272$)
Polite forms	19% ($N = 9$)	20% ($N = 30$)	0.05	6% ($N = 10$)	7% ($N = 38$)
Presuppositions	67% ($N = 32$)	50% ($N = 75$)	3.90*	40% ($N = 64$)	24% ($N = 128$)
Reference to own emotions	6% ($N = 3$)	21% ($N = 31$)	5.38*	2% ($N = 4$)	9% ($N = 50$)
Reference to own feelings	15% ($N = 7$)	36% ($N = 54$)	7.96**	% ($N = 10$)	% ($N = 85$)
Self-disclosure	6% ($N = 3$)	25% ($N = 37$)	7.75**	2% ($N = 3$)	10% ($N = 55$)
Strong assertions	48% ($N = 23$)	15% ($N = 23$)	21.39***	19% ($N = 30$)	6% ($N = 30$)
Supporting statements	27% ($N = 13$)	42% ($N = 63$)	3.54	11% ($N = 17$)	21% ($N = 114$)

* Significant at $p < 0.05$.
** Significant at $p < 0.01$.
*** Significant at $p < 0.001$.

that 10 of the 16 stylistic codes were found to reveal a significant difference in terms of male and female usage. A significantly higher proportion of females made contributions containing agreement, $\chi^2(1, N = 197) = 11.30$, $p < 0.01$, empathy, $\chi^2(1, N = 197) = 4.91$, $p < 0.05$, personal experience, $\chi^2(1,\ 197) = 4.41$, $p < 0.05$, and self-disclosure, $\chi^2(1, N = 197) = 7.75$, $p < 0.01$, in comparison to males. Furthermore, significantly more females made reference to their own emotions, $\chi^2(1, N = 197) = 5.38$, $p < 0.01$, and own feelings, $\chi^2(1, N = 197) = 7.96$, $p < 0.01$, in comparison to males. On the other hand, a significantly higher proportion of males made challenging, $\chi^2(1, N = 197) = 19.59$, $p < 0.001$, and controversial contributions, $\chi^2(1, N = 197) = 14.14$, $p < 0.001$, expressed disagreement, $\chi^2(1, N = 197) = 17.86$, $p < 0.05$, used humour, $\chi^2(1, N = 197) = 19.74$, $p < 0.001$, and presuppositions, $\chi^2(1, N = 197) = 3.90$, $p < 0.05$, and strong assertions, $\chi^2(1, N = 197) = 21.39$, $p < 0.001$, in their postings.

The results of the supercode analysis are shown in Table 3. The supercode queries are listed, as are the corresponding proportions of males and females making each type of contribution, on which χ^2 analyses were conducted. The percentages of each type of posting, expressed as a percentage of the total male and female postings are also given in the table. Significantly more females made contributions that were attenuated but not authoritative, $\chi^2(1, N = 197) = 16.26$, $p < 0.001$, and consisted of female language features only, $\chi^2(1, N = 197) = 19.43$, $p < 0.001$, than males. Significantly more males made contributions that were authoritative but not attenuated, $\chi^2(1, N = 197) = 14.88$, $p < 0.001$, and consisted of male language features only, $\chi^2\ (1, N = 197) = 11.50$, $p < 0.001$, than females. There were no significant gender differences found along the task-orientated versus socioemotional dimensions. However, the socioemotional code family was further divided into positive socioemotional and negative socioemotional families. This revealed that significantly more females engaged in positive socioemotional behaviour than males,

Table 3
Interaction style supercode analysis by gender: χ^2 of male and female samples

Supercode	% of participants		χ^2	% of postings	
	Male ($N=48$)	Female ($N=149$)		Male ($N=161$)	Female ($N=538$)
Attenuated not authoritative	31% ($N=15$)	64% ($N=96$)	16.26**	16% ($N=26$)	41% ($N=218$)
Authoritative not attenuated	69% ($N=33$)	37% ($N=55$)	14.88**	40% ($N=64$)	16% ($N=85$)
Male not female language	33% ($N=16$)	12% ($N=18$)	11.50**	12% ($N=20$)	4% ($N=19$)
Female not male language	33% ($N=16$)	69% ($N=103$)	19.43**	20% ($N=32$)	44% ($N=235$)
Male and female language	73% ($N=35$)	65% ($N=97$)	1.00	61% ($N=98$)	46% ($N=246$)
Positive not negative	35% ($N=17$)	58% ($N=86$)	7.24*	19% ($N=30$)	35% ($N=190$)
Negative not positive	56% ($N=27$)	19% ($N=28$)	25.32**	27% ($N=44$)	7% ($N=38$)
Task-oriented not socioemotional	56% ($N=27$)	63% ($N=94$)	0.72	36% ($N=58$)	30% ($N=161$)
Socioemotional not task-oriented	33% ($N=16$)	37% ($N=55$)	0.21	16% ($N=25$)	13% ($N=71$)
Task oriented and socioemotional	65% ($N=31$)	73% ($N=109$)	1.29	44% ($N=71$)	52% ($N=279$)

* Significant at $p<0.01$.
** Significant at $p<0.001$.

$\chi^2(1, N=197)=7.24$, $p<0.01$, whereas significantly more males engaged in negative socioemotional behaviour than females, $\chi^2(1, N=197)=25.32$, $p<0.001$.

4. Discussion

This research investigated the existence of gender-related patterns in language use and interaction styles in a formal, educational context. Firstly, participants' use of individual linguistic variables was examined, such as absolute adverbials (e.g., 'obviously') and references to emotion (e.g., 'love'). Only one significant difference was found between males and females regarding use of the 12 linguistic variables. More females used intensifiers such as 'really' and 'totally' in their postings. This does not support the work of Herring (1993) or Savicki et al. (1996) who reported gender differences in use of individual linguistic features in online contexts, such as first person and plural pronouns, absolute adverbials, interjections and qualifiers.

In this context, first person pronoun usage did not vary by gender. The electronic discourse showed extensive first person pronoun usage and 87% of postings were of this nature. Also, over a third of postings expressed agreement or disagreement with other students. Just over a fifth of all student postings contained references to own emotions or self-disclosure. These findings in this formal context suggest that the computer-mediated context is perhaps not as impersonal as previously implied (e.g., Markus, 1994; Sproull & Kiesler, 1986) and provides support for social information processing theory (Walther, 1992) in that CMC can support socioemotional communication. However, the discourse was also characterised by a lack of emoticons. This could be due to the formal, asynchronous context or a lack of participants' experience with CMC. Alternatively, participants may have felt that they were able to convey meaning adequately through their language use and did not have to resort to using emoticons to convey the tone of their message. There was some evidence of the use of paralanguage to convey tone, although this was relatively rare.

Although use of only one linguistic variable was found to significantly differ by gender, 11 out of the 16 stylistic variables produced significant results. It was found that significantly more females made contributions containing empathic utterances, personal experience, self-disclosure, references to own emotion and references to own feelings, than males. On the other hand, more males than females sent postings containing controversial statements, humour, strong assertions and presuppositions.

The 'supercode' analysis in Atlas.ti 4.2, which examined use of combinations of language and interaction styles, showed that females were more likely to make attenuated contributions and use only traditional female language features in their postings. On the other hand, males were more likely to make authoritative postings and use only male language features. These findings provide empirical support for the existence of gendered styles of communication, as opposed to the influence of gender on use of individual language features. They also support the communication styles identified by Herring (1994) to an extent. However, they suggest differences in the degree to which gendered discourse styles are utilised, as opposed to differences in kind. The male style that Herring (1994) described was a considerably more extreme authoritative and adversarial style that included sarcasm and flaming, which did not occur often in the present research. This could have been due to the differences in context, in that Herring (1994) examined public discussion lists and the present research was concerned with private discussion forums in which instructors were present.

However, the results do support Herring's work in that females were found to employ personal and emotional forms of language more than males, who in turn used more authoritative language. It is possible that participants, especially females, developed ways of overcoming what could be perceived as an impersonal environment through the development of a discursive norm of explicitly personal writing. For example, the tendency to disclose personal information beyond that of opinions and feelings. It should be noted here that the coding category of self-disclosure was taken to mean high self-disclosure in the form of personal information beyond that of thoughts, feelings and experiences, which were distinct coding categories in themselves. It is interesting to note that Herring's prediction that the minority gender group will accommodate to the style of the majority was not supported in that gender differences were found in interaction style and males did not generally conform to the female norm of explicitly personal writing.

The findings here support Matheson and Zanna's (1990) claims that CMC users are more likely to experience high-private self-awareness, as opposed to deindividuation. High levels of first person pronoun use were observed and participants disclosed personal opinions, feelings, emotions and experiences, which may indicate a high level of private self-awareness. It is interesting to note that the pseudonymity offered in this research was not found to affect the level of self-disclosure, as similar proportions of pseudonymous and real name participants self-disclosed. However, gender was found to significantly influence the level of self-disclosure online as more females self-disclosed than males.

Therefore, previous work in the area of gender and language, both offline and online, is supported to some extent. Herring (1994), Savicki et al. (1996) and Soukup (1999) found that males used more authoritative language than females and that more females revealed thoughts and feelings than males. However, Michaelson and Pohl (2001) did not report any gender differences along the supportive/emotional versus adversarial/task-oriented dimension in their study of CMC interactions. Although this study did support the supportive/emotional versus adversarial distinction to an extent, it did not support the task-orientated versus socioemotional distinction in relation to gender. It is argued here that this distinction is too simplistic. Males and females were just as likely to send messages of each type and, more commonly, messages containing both task-orientated and socioemotional content. The distinction between these categories was problematic as participants were frequently engaging in task behaviours due to the specific formal purpose, some of which could also be classed as socioemotional behaviour (e.g., requests opinions).

Furthermore, as indicated in the introduction, definitions appear to focus on the positive elements of socioemotional discourse such as expressions of support. However, disagreement and challenging utterances were also categorised as socioemotional discourse in this research, as they also involved references to others and reacting to others. Thus, a distinction was made between positive and negative socioemotional behaviour. Gender-related patterns were found as males had a tendency to post negative responses, whereas females were more likely to respond positively to other participants, with expressions of agreement and similarity in opinion and experience.

The finding that males tend to express disagreement more, whereas females tend to express agreement, supports Coates (1993) and Tannen (1991) that the male style is based on competitiveness and the female style is based on cooperativeness. It could also reflect gender-related learning preferences, as females may prefer to learn through connectedness and cooperativeness, whereas males may prefer a more independent and argumentative learning environment (see Belenky, Clinchy, Goldberger, & Tarule, 1997; Gilligan, 1992). These differing epistemologies and communication styles may influence the sharing and acquisition of knowledge in mixed-gender, computer-mediated educational interactions. For example, the increased tendency for females to agree and support, as opposed to disagreeing and rejecting suggestions, could be interpreted as taking a low-power role in the discussion. Tannen (1991) states that women typically use more supportive language patterns, which have the effect of diminishing the power of their own contributions. Thus, some participants, particularly males, may devalue and potentially ignore contributions of a personal and attenuated nature.

Furthermore, Taylor (1978) noted that women are often criticised for using language that is too personal. It is presumed that such writing blunts the 'argumentative edge', which is a conventional hallmark of effectual exposition. The issue of the perceived credibility of a contribution that uses personal and attenuated language is worthy of further investigation, as these linguistic forms are often associated with negative stereotyping, which may affect the perceived power and persuasiveness of the communication. For example, an authoritative and challenging contribution may be perceived as a more convincing argument and a more valuable contribution to knowledge than a personal and cooperative one by other participants and perhaps even instructors, who may be assessing the quality of online contributions.

The emerging female discursive norm of self-disclosure is likely to have been influenced by the nature of some of the psychological topics under discussion, which appeared to be personally relevant to some participants (e.g., eating disorders, depression). Indeed, Wallace (1999, p. 151) argued that the 'tendency to disclose more to a computer…is an important ingredient of what seems to be happening on the Internet'. Therefore, an increased tendency to self-disclose online, combined with the discussion of potentially sensitive topics, raises an ethical issue for the use of CMC in psychology courses, which instructors should be aware of.

It is also recommended that instructors should take the gender differences in communication styles identified in this research into account when using computer-mediated discussion forums in education, particularly when establishing 'ground rules' for electronic discourse and producing guidelines for the use of such forums. There is the possibility that some CMC participants could be intimidated by contributions that use an authoritative style to critique ideas and express disagreement and find them unconstructive. Indeed, Herring (1994) suggested that differences in communication ethics between males and females regarding what is both acceptable and credible forms of communication are likely to result in male dominance and female submission in CMC. However, this perhaps implies a conscious and ethical choice to post in a particular style, rather than a reflection of a subconscious gender identity. Further research is needed to investigate this issue.

It is possible that the gender-related patterns reported here, which are similar to traditional gender role stereotypes, were found as the salience of gender as a social category may have invoked norms and stereotypical expectations regarding gender appropriate behaviour (Matheson & Zanna, 1990; Reicher et al., 1995). Gender was made a salient social category in the online discussion to an extent, as threads developed surrounding the issue of gender in relation to the topics of eating disorders, abortion and even task behaviour. Moreover, some participants used their real name online and the tendency for females to opt for a numerical identification more than males could perhaps have acted as a gender marker in itself. It was also found that some participants revealed their gender in the content of their contributions (e.g., 'as a gay guy', 'as a 19-year-old girl'), showing that pseudonymous conditions do not necessarily prevent gender from being revealed online.

Matheson (1991) suggested that high private self-awareness would facilitate access to internalised frameworks for processing and responding to social cues such as gender. Therefore, the combination of high private self-awareness and the salience of gender cues may lead to expectations of gender-appropriate behaviour based on traditional stereotypes. These expectations could promote self-fulfilling prophecies, as Snyder (1984) suggested, resulting in stereotypically gendered behaviour. Therefore, this could account for the gender-related patterns in language use and interaction style found in the present research. For example, it could be the case that more females were found to self-disclose than males because the salience of gender as social category may have invoked behavioural norms and expectations based on stereotypes regarding gender appropriate behaviour, effectively increasing the level of self-disclosure by females and lowering the level of self-disclosure by males.

The notion that gender does not matter online seems overly optimistic, especially considering that one of the most popular first questions asked in recreational chat rooms is 'asl', which stands for 'age, sex, location?'. Internet users use this initial question to get a sense of who they are talking to and sex (or gender) is one of the major social categories that are considered, often making it a salient social category in online contexts. For example, Huffaker and Calvert (2005) found that both males and females often reveal personal information online such as their real names, ages and locations.

Finally, the unequal male and female sample sizes were unavoidable in this context as female students often outnumber male students on psychology modules and it is argued that studying gender and CMC in a meaningful context outweighs this potential drawback. Future research should investigate the extent to which the gender-related patterns found here can be extrapolated to other educational computer-mediated contexts, for example male-majority courses such as engineering. The relationship between gender identity and expression and manipulation of that identity in online text-based contexts is also deemed worthy of attention.

In conclusion, the results provide support for gendered styles of communication online, showing that CMC does not guarantee a gender-free environment. Cues to gender were found to exist in the language used by CMC participants and as the work by Thomson and Murachver (2001) suggests, CMC users may be able to identify gender based on these cues alone. The gender gap may be closing in terms of being online, although women still remain less frequent users of the Internet (Ono & Zavodny, 2003). It is likely however that different results will be found depending on the context in which Internet use is being studied. For example, Miller and Durndell (2004) found evidence for equal participation in the context of online discussion groups in education. However, the present study indicates that even within the educational domain, if you scratch beneath the surface, gender differences in language use and interaction style can be revealed. This suggests that existing gender-related patterns of power and communication may carry over into online environments and compromise the extent to which an equitable environment can be maintained. Therefore, it seems that gender does matter online.

Appendix A. Descriptions and examples of linguistic variables

Code	Description	Examples from corpora
Absolute adverbial	Strong assertion	'obviously', 'certainly'
Exhortation	Phrased as suggestion	'let's', 'why don't we'
First person plural pronoun	Refer to group including writer	'our', 'we', 'us'
First person pronoun	Refer to writer	'me', 'my', 'I', 'myself'
Second person address	Refers to another group member	'you', 'your'
Imperative verb	Expresses a command	'think about it'
Impersonal truths	Presupposed fact	'it is a fact that'
Intensifier	Reinforces meaning upwards or downwards on a scale of intensity	'so', 'really', 'very', 'totally', 'quite', 'hardly'
Interjection	Exclamation, usually emotive in meaning	'ah!', 'oh!', 'eh?', 'oops', 'well', 'boy!'
Qualifier/hedge/tag question	Word or phrase that modifies the meaning of another word or phrase	'perhaps', 'may', 'seems', 'sort of', 'isn't it?'
Reference to emotion	Use of emotive word	'happy', 'depressing'
Rhetorical question	Assertive question not meant to be taken literally	'but isn't this a 2-way street?'

Appendix B. Descriptions and examples of stylistic variables

Code	Description	Examples from corpora
Agreement	Makes explicit statement of agreement	'I totally agree with you'
Challenging	Statements that challenge others' ideas or opinions	'How can you possibly think that'
Compliments	Author praises participant for contribution	'I thought the points you made were very good'

Appendix B (*continued*)

Code	Description	Examples from corpora
Controversial	Makes controversial statement, statements likely to 'wind people up'	'People without the intellectual capacity should be discriminated against'
Disagreement	Makes explicit statement of disagreement	'I don't agree with you'
Empathic	Shows empathy (sense and understand someone else's feelings as if they were one's own)	'They have a lot of pressure on them'
Humour	Quality of being amusing	'Unless you wanted to be an accountant, which is crazy anyway!'
Personal experience	Refers to speaker's own experience	'When I was at school'
Personal orientation	Refers to speaker's personal orientation	'In my opinion', 'I think that'
Polite forms	Polite forms of language such as thanking and apologising	'Thankyou for your response', 'I'm sorry'
Presuppositions	Statements that assume certain facts or opinions	'People would end up as robots'
Reference to own emotion	Makes reference to speaker's own emotions	'It makes me sad'
Reference to own feelings	Makes reference to speaker's feelings	'I am very interested in'
Self-disclosure	Shares personal information beyond that of opinion, experience and emotion	'I am bulimic'
Strong assertions	Assertions made without or with involved modalities	'I am very sure that'
Supporting statements	Statements that support a previous view or argument (with or without an explicit statement of agreement)	'(I agree) There is no way of being certain that cloned scientists or nobel prize winners will have good intentions'

References

Barrett, E., & Lally, V. (1999). Gender differences in an on-line learning environment. *Journal of Computer Assisted Learning, 15*, 48–60.

Belenky, M. F., Clinchy, B. M., Goldberger, N. R., & Tarule, J. M. (1997). *Women's ways of knowing*. New York: Basic Books.

Biggs, J. (1987). *Student approaches to learning and studying*. Hawthorn, Victoria: Australian Council for Educational Research.

國立雲林科技大學
96 學年度博士班招生入學考試試題

系所別：資訊管理系
科　目：資訊科技文獻

第一部份：

Automatic Management of Laboratory Work in Mass Computer Engineering Courses

1. 這一題的目的是測驗你的文字表達能力。請以中文撰寫本篇論文的摘要。 15%
2. 這一題的目的是測驗你撰寫技術論文的能力。請以中文及圖形（如流程圖）說明 Automatic Project Evaluator 的設計。 20%
3. 請描述一個軟體的設計，這個軟體能夠幫助教師以自動化程度高的方式，製作論文中提到的練習題目的 test definitions。 15%

第二部份：

Envisioning Intelligent Information Technologies through the Prism of Web Intelligence

1. 作者認為處理大型複雜的實務問題的關鍵牽涉至少三種方法論。請說明該三種方法論。(限以中文作答) 10%
2. 智慧型企業入口網站之核心技術是什麼，該技術應該提供哪些功能？(限以中文作答) 15%
3. 請描述本篇論文的主要論點。(限以中文作答) 10%
4. 請評論此篇論文。(限以中文作答) 15%

Automatic Management of Laboratory Work in Mass Computer Engineering Courses

Antonio García, Santiago Rodríguez, Francisco Rosales, and José L. Pedraza

Abstract—Teaching computer engineering calls for an important practical component, usually covered by setting several laboratory exercises for each course. These exercises are specified as assignments by the teachers and have to be completed by the students. At the Computer Science School of the Technical University of Madrid (UPM), Madrid, Spain, some of these laboratory exercises have to be set for up to 400 students. High-quality laboratory work requires the use of technology to help in student management, interaction, and assessment. Over the last ten years, this department at UPM has been a site for new tool development. Students use these tools to submit their laboratory work over the network. These tools allow the students' work to be checked by a battery of tests proposed by the instructor. In addition, a specific tool has been built to detect plagiarism by students. This paper describes the services provided by this environment and the experience gathered during the use of this laboratory.

Index Terms—Automatic assessment, computer science/ engineering learning, interactive learning, mass courses techniques, plagiarism detection.

I. Introduction

COMPUTER architecture is one of the core subjects that both computer science and computer engineering students have to learn. Specifically, students have to learn the principles of how a computer works, what its components and connections are, and the basis of assembly language programming. Because of the strong relations between all the computer components, this body of knowledge is difficult to acquire.

Teaching in this area involves both a theoretical and a practical part. Practical work is needed to reinforce the subjects explained in the theoretical part. This reinforcement can be achieved by setting a specific problem to be solved by students. This practical work allows students to apply the previously acquired knowledge. It also helps them to clarify any misconceptions or doubts by asking teachers for additional information and explanations.

Unfortunately, there are some situations in which this teaching approach cannot be directly applied. For instance, it is not adequate in some educational environments, such as distance education or in mass courses, where the student/teacher ratio is relatively high. In both contexts, the traditional relationship between students and their teacher is necessarily not as close. Several approaches for developing new environments to provide distance learning and automatic evaluation, and for conducting laboratory experiments, have been considered. For example, [1] shows an environment to share theoretical material at a distance education university. Another use of Web technology that obviates classrooms by allowing asynchronous interaction between students and the instructor is described in [2]. Some attempts have been made in the area of automatic assessment, but always focusing on specific topics. For instance, [3] and [4] present two systems for filling Web-based test questionnaires.

Generally, when considering the introduction of laboratory work in distance education, new problems have to be solved. The time spent by teachers on settling student doubts will increase, especially if the number of students is high. Laboratory work will also be harder to evaluate than theoretical material. The TRAKLA[1] system, shown in [5]–[7], is an environment for automatically evaluating small programming exercises. However, it does not seem to be usable for larger exercises or for small projects.

There are several motivations for proposing small projects as the basis of the practical part of engineering courses.

1) Any such course involves many interrelated concepts.
2) Setting small projects as laboratory work brings students closer to real-world working practices. In this case, each proposal should be a small self-contained project described in a specification document delivered to the students.
3) Small projects avoid otherwise needed exercise personalization. Teachers are aware that the shorter the exercise, the less freedom students have. When short exercises are proposed, especially in distance education, the solutions provided by different students when personalization is allowed will result in nonidentical solutions.

Consequently, students are asked to develop a few small projects instead of numerous exercises.

At the Computer Architecture Department of the Technical University of Madrid (UPM), Madrid, Spain, instructors teach several topics to about 400 students per course. Most students have to do at least one of the laboratory exercises during their second, third, and fourth years. This environment clearly falls into the class of mass courses, which led this department to tackle the problem in 1994. At that time, a set of tools was being developed, allowing students to get information automatically about the achievement and the overall quality of their respective laboratory exercises. In this environment, teachers should

Manuscript received July 21, 2003. This work was supported in part by the Spanish research agency under Project TIC-2003-08933-C02-02.

The authors are with the Department of Computer System Architecture and Technology (DATSI), Technical University of Madrid, 28660 Madrid, Spain (e-mail: dopico@fi.upm.es; srodri@fi.upm.es; frosal@fi.upm.es; pedraza@fi.upm.es).

Digital Object Identifier 10.1109/TE.2004.832874

[1]The name comes from the Finnish acronym "TietoraKentect ja Algoritmit; KotiLaskujen Arvostelu," which means "Data Structures and Algorithms; Homework Exercises Assessment."

0018-9359/$20.00 © 2005 IEEE

act more like advisors than lecturers. Teachers provide students with guidance for understanding and addressing the specified problem and focus their efforts on solving the problems they encounter. Another important job for the teacher is to provide students with feedback to help them learn working methods.

These proposed laboratory exercises help students to learn different computer-architecture-related and operating-system-related topics. These topics range from computer arithmetic to the development of a reduced shell interpreter. They include management of data structures in assembly language, input/output (I/O) techniques, interrupts and exceptions, microprogramming, and shell script programming. Each of the laboratory exercises proposed is a small self-contained project described in a specification document delivered to the students. Students are provided with this document, and they have to understand it fully to pass this part of the subject.

Students are notified in advance of the assessment criteria. Their laboratory work will be checked against a battery of tests that it has to pass. Manual checking of each student's laboratory exercise is not possible because it would take an unreasonable amount of time. Moreover, all students would have to be allowed to interact with the teacher to correct the errors before being given another chance.

In 1993, before this group started to develop the tools described in this paper, the authors gathered a considerable amount of statistical information about the laboratory work performed by the students. From the analysis of this data, some weaknesses in student laboratory work results were observed, especially on three points.

1) The definition of the practical work was subject to no constraints. This lack of constraints sometimes led teachers to provide an incomplete specification. Then, students did not fully understand what they had to do. In the worst case, some students were even unable to solve the set exercise.
2) Students did not think enough about the construction of a battery of tests to check properly the results of the laboratory work themselves.
3) Students did not debug the code. As a consequence, they did not acquire the expected experience on how to debug programs.

The authors decided to get involved in the definition and development of automatic management tools for computer science and engineering laboratory work environments. The only alternative to this kind of automatic management tools would have been to reduce the number or the depth of some or all of the laboratory exercises. Unfortunately, this solution would mean that students would not acquire the knowledge that they were expected to have in the long term.

In this paper, the authors describe in detail a new set of tools that makes use of the Worldwide Web for remote education and automatic assessment. Even though some work has been done in the field of distance education for mass courses, no other tool allowing the management of small projects has been reported. The objective of these tools is to help educators to provide a good laboratory education experience without consuming too many resources.

II. Computer Architecture and Operating Systems Laboratory Work

Computer architecture covers many conceptual topics that are difficult to learn without performing some practical work. The proposed laboratory work concerns the following aspects:

- *Assembly language programming*: The objective is to acquire hands-on knowledge of the actual programming language understood by the computer. An emulator of the easy-to-program mc88110 Reduced Instruction Set Computers (RISC) processor described in [8], [9] is used.
- *Microprogramming*: The objective is to get a global view of internal computer operation and of the connections between its functional units.
- *I/O system*: The student will do this work by programming two interconnected serial ports under different simulated conditions using interrupts. An emulator of the mc68000 processor [10] is used.
- *Cache memory*: The objective is to experiment with the basis of hierarchical memory systems, particularly cache memories, their fundamentals, and how they work. This exercise is carried out using the same mc88110 emulator as mentioned previously.

The objective within the operating system subjects is for the students to learn how the operating system is designed and how to use its services. The proposed laboratory exercises are listed hereafter.

- *C programming language*: Students will develop programs using this extended language, which will be needed in other laboratory exercises.
- *Command shell*: Students have to design and code a command interpreter. This project ensures that students interact with the most important operating system services.
- *Kernel additions*: Students have to modify an open-source operating system kernel (Linux) to provide it with extended functionality. They also have to analyze the memory manager behavior.

Each of the laboratory exercises briefly described previously is a self-contained project to be completed by the student. These projects cannot be personalized, primarily because of the number of students taking the course. In such situations, personalization would require considerable staff power, which is not usually available.

III. Laboratory Work Environment

An important problem found in environments in which a large number of students have to complete small projects, or even exercises, is their evaluation, as [7] shows. This problem results from the large number of human resources needed, making evaluation impractical. The only affordable way to tackle this problem is to introduce automatic project management and evaluation tools.

The authors developed several such tools to assist teachers. Students make use of these tools by means of a standard Web interface. The most important starting points considered during the design refer to the need to make material and instructors available to students, as explained hereafter.

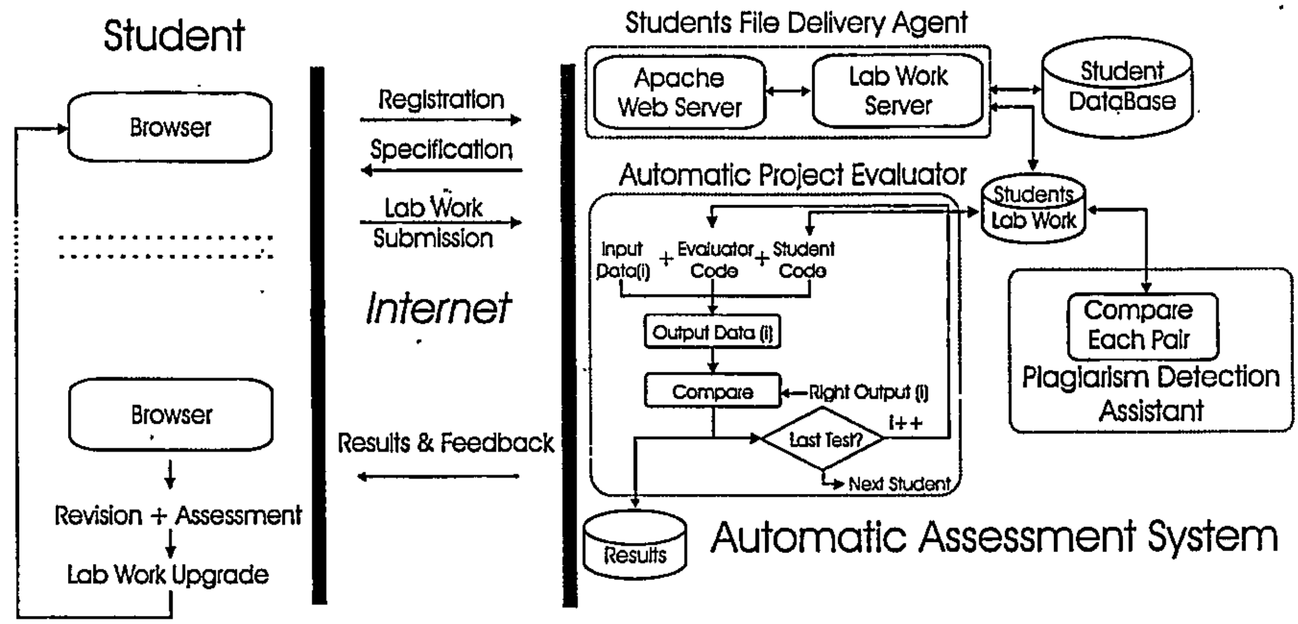


Fig. 1. Laboratory work automation environment.

- *Material accessibility*: Project specifications can be downloaded from the department Website and also be available as printed matter.
- *Instructor accessibility*: Students can ask teachers for help, both by e-mail or personally. The students are encouraged to use e-mail because it saves time. This practice makes students more conscious of the importance of writing good descriptions of their problems. It also contributes to the development of a frequently asked questions (FAQ) file containing student questions and teacher explanations that will be later consulted by other students coming across the same kind of problems.

The main set of tools forming this environment is listed as follows:

1) *Student files delivery agent*: Students use this software tool to send electronic versions of their completed work to the instructor responsible for the laboratory exercise. When their projects have been automatically evaluated, they can also use this same delivery tool to get the results of their evaluation. The student can use this tool at any time during the period assigned to a specific laboratory exercise.
2) *Automatic project evaluator*: This tool has to be configured initially, and a maximum number of available evaluations per student must be defined. Once configured, this tool executes a set of achievement and quality tests for each student laboratory exercise entitled to evaluation. It determines which tests the work passes and which ones it fails, giving an indication of the reasons for each failure. Students can use the delivery agent to get the results obtained by their laboratory work after this evaluation. These results usually give them enough feedback to revise, think again, and correct their laboratory work. In this way, they can learn from their own mistakes. This tool can be started as either an immediate response to the reception of a student laboratory exercise, or alternatively, it can be executed at specific times during the period in which students are developing their projects.
3) *Plagiarism detection assistant*: An unpleasant but important problem is plagiarism among students completing the same laboratory work. This problem is almost directly derived from the large number of students enrolled at the same time in the same course. Some students have the wrong idea about the possibility of their unauthorized copies being detected. The plagiarism detection assistant compares each laboratory work against all the others, giving an estimation of their similarities. Analyzing the most similar pairs, teachers can locate plagiarism, if any, and act accordingly.

The following sections describe the main aspects of the three modules forming the core of this laboratory work automation environment: student files delivery agent, automatic project evaluator, and plagiarism detection assistant. The relation among different components of the environment is shown in Fig. 1.

A. Students File Delivery Agent

The delivery agent module consists of a client and a server program. They jointly allow students to establish contact, from any Internet-access-equipped computer, with the department machine used as the host of the laboratory work automation environment. The client can be a small and specific program provided to the students, or it can be a conventional Web browser.

The server is permanently running on one of the department hosts. It is configured for each laboratory exercise. One aspect of the configuration of this server is to provide the list of all students allowed to carry out the laboratory work in question. Only the students on the list can successfully interact with the server. This security option prevents undesired accesses. How-

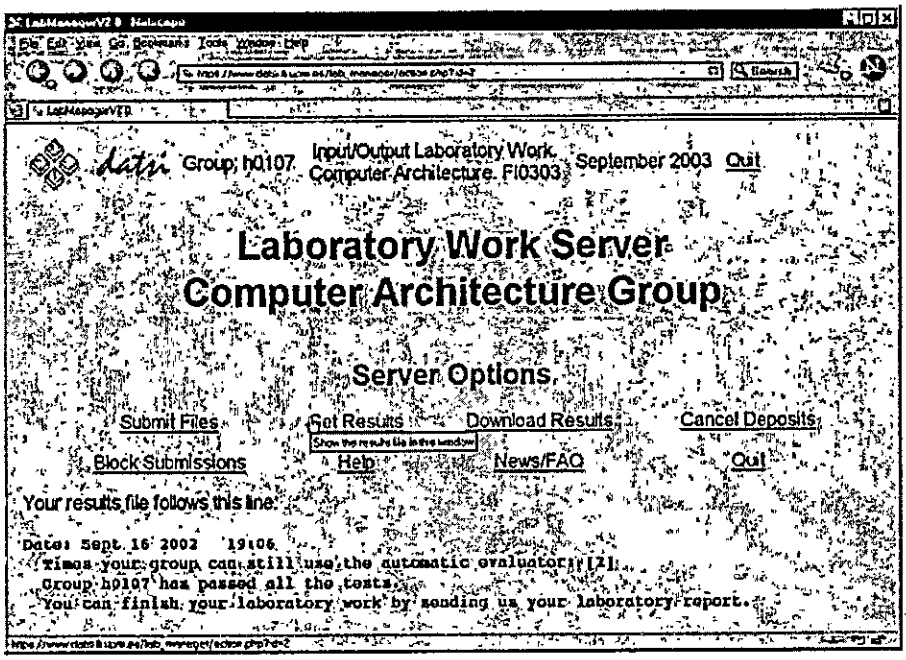


Fig. 2. Laboratory environment: Web interface.

ever, more flexible access is sometimes convenient and, if so, can be provided by creating a generic student entry in the list.

Once the student has established contact with the server, several actions can be taken by selecting the appropriate menu option. Available options for one of the courses are shown in Fig. 2, representing the user interface available for the students.

In Fig. 2, the first option allows the student or group[2] to send its locally stored version of the previously established files. The second option lets the student view the evaluation results of previously submitted files in the same window. This option is shown in Fig. 2 with a tool tip in front of it. The next option lets the student download the evaluation results file. The student connected to this Web page can also cancel the automatic evaluation of a previously delivered set of files (a *deposit*). This option has been made available because in nearly all of the laboratory work, only a limited number of automatic evaluations per student is allowed. If the students find an error in their files and prefer not to have their work evaluated yet, they can make use of this option. On the other hand, if students get an acceptable result from the system (for instance, all the tests are passed), they can block further deposits on their account. This option cannot be changed; therefore, if it is selected by mistake, the student should contact the teacher directly. The purpose of the last three options available on this page is clearly indicated by their respective menu titles: get help, get the latest general news about the laboratory exercise, and terminate the session with the server. This last option is provided for security reasons to prevent other users from interacting with the system on behalf of the student who has just finished his or her session.

[2]Students are allowed to do the work necessary to complete their projects in groups of no more than two or three people. From now on, the term "student" will be used to refer to either an individual student or a group of students collaborating on the same exercise.

In this department, every laboratory exercise has an instructor who is responsible for its management. The delivery agent provides this instructor with several tools or utilities. One such utility is the continuous generation of a log file in which all the established connections are registered. Every student connection adds several lines to this log file, including the login name, the name or ip-address of its local computer, and the time at which each action was taken. A fragment of a log file is shown in Fig. 3.

Besides the global log file, the laboratory work automation environment stores a copy of all the files provided by each registered student and the respective result file obtained during the evaluation, including files whose previous deposit was already canceled.

There are several reasons why all this information is stored about every student history. Probably the most important one is to allow both the teacher and students to solve possible minor conflicts when they hold different viewpoints about specific facts related to the work evaluation. Finally, all the available information that can be taken into consideration is very important when a student is suspected of plagiarism.

B. Automatic Project Evaluator

Mass courses requiring laboratory exercises would need a sizeable number of human resources to help students achieve the objectives. This problem was tackled using the constructivism paradigm for laboratory work [5], [11]. The main idea is to get students to learn by themselves and to build up their knowledge. This approach, as opposed to objectivism, requires major changes in how teachers and students understand the teaching process. The teacher is considered to be an advisor, who must be available to solve the problems encountered by the students when they are solving the laboratory exercise.

```
Nov 04, 2002 12:33 f980769  REGISTERING (OK) [batman.fi.upm.es]
Nov 05, 2002 09:30 a920157  LOGIN (ERROR) Invalid Password [batman.fi.upm.es]
Nov 05, 2002 09:31 a920157  LOGIN (OK) [batman.fi.upm.es]
Nov 05, 2002 09:31 a920157  NEWS (OK)
Nov 05, 2002 09:32 a920157  DEPOSIT (ERROR). File buffer.s: Invalid format error
Nov 05, 2002 09:32 a920157  DEPOSIT CANCEL (OK) Submission error
Nov 05, 2002 14:13 960016   LOGOUT (OK)
Nov 05, 2002 14:14 960016   LOGIN (OK) [batman.fi.upm.es]
Nov 05, 2002 14:14 960016   DEPOSIT (OK) [batman.fi.upm.es]
Nov 05, 2002 14:14 960016   LOGOUT (OK) [batman.fi.upm.es]
Nov 05, 2002 14:14 a0343    REGISTERING (ERROR) Unknown authors [212.166.139.183]
```

Fig. 3. Global log file fragment.

This schema works fine in small groups, where the teacher is quite close to students, but this approach does not appear valid for large courses. Obviously, if this approach is to be applied to mass courses, some kind of automatic laboratory work management needs to be introduced to help teachers to receive and assess student work. In addition, it would provide individual information for each student about the quality of his or her laboratory work.

The automatic project evaluator is a tool for helping teachers to manage mass laboratory work courses and also for helping students to achieve their objectives. It works from a set of predefined tests, which are executed for every laboratory exercise of every student. For each exercise, it writes a text file recording which tests the exercise successfully passed and which tests it failed. Afterwards, students can access this file using the delivery agent and be informed about the quality of their laboratory work.

The objectives set for students regarding the laboratory exercises can be summarized by considering that after finishing the laboratory exercise, they should be able to perform the following tasks:

- to interpret the specification and to design a program according to this specification;
- to build their own set of tests to check their own work;
- to debug their program;
- to work in collaborative environments with other colleagues.

To achieve these objectives, students can request to try a test a limited number of times, established by the teacher. Each time a student asks to do a test, his or her laboratory work is executed with several input data sets established by the teacher. These data sets are unknown to students. The limitation on the number of tests available for each student is imposed to prevent them from using the evaluator as a debugger, ruling out trial and error. In this way, students are encouraged to debug their laboratory work to assure, as far as possible, that it works as expected. To carry out this debugging, they need to design and build a set of tests according to the specification provided. They will execute their laboratory work with their own input data sets because they do not know the input data set established by teachers.

The evaluator is executed at predefined daily times known to the students. Every time students want to check their laboratory work, they have to submit the files using the delivery agent provided by the department. In this way, students will be automatically registered by the system, and their laboratory work will be tested the next time the evaluator is activated, usually overnight.

Every laboratory assignment to be automatically evaluated has to be configured for a set of tests prepared by the teacher. Every test definition consists of the following four elements.

1) *Configuration file*: It defines the data set to be analyzed once the test has been run: the subset of the processor registers, several memory addresses, etc.
2) *Test input data file*: It contains both the calling program and the input data set that has to be provided for the test.
3) *Correct results file*: This file contains the results that the teacher expects to get from the execution of the student laboratory work.
4) *Error message file*: It contains the text message to be provided to the student if the test is not accepted by the evaluator. This message will contain both predefined and run-time-generated information that will help the student to find the error.

The set of tests established to evaluate laboratory work should cover the whole specification provided by the teacher. Each individual test can be assigned a given level so that there may be compulsory and optional tests. If a compulsory test is not successfully passed, then the complete laboratory work cannot be passed. However, some optional, usually more complex tests, are used to determine the student's final grade.

When students retrieve the results file, they have to look for errors. The description of the error found in the student laboratory work depends on how complex the associated test is. Generally, a description of the test is reported, and it suggests the type of the error found—for example, an infinite loop, a memory content error, or an undesired exception. For example, whenever a memory buffer contains the wrong results for a test, the evaluator provides the student with a description of the error. It includes both the wrong data found and the correct data that should be there. Students try to fix the described errors using this feedback. If they are unable to locate the error, they can ask the teacher for help personally or by e-mail. Experience shows that most of the questions answered by the teacher are related to misconceptions. For example, teachers answered 190 e-mail questions in the last I/O laboratory work. Of these, 70% were related to misconceptions about the basis of I/O; less than 1% were a result of laboratory work documentation problems; and the remainder were results of several aspects, including course organization. The small number of documentation-related questions is worth mentioning.

C. *Plagiarism Detection Assistant*

The number of students having to complete each laboratory exercise has another unfortunate consequence; it raises the chances of plagiarism. Because of the number of students concerned, some of them think plagiarism is impossible to detect, as described in [12]. To detect students trying to exploit other people's efforts to complete their exercises, a plagiarism detector assistant has been added to the environment.

The plagiarism detector runs on a department computer independently of the evaluator, and it can be run before or after the exercise has been evaluated. This tool makes an exhaustive search comparing each student file set with all the other ones, including similar exercises from previous years. According to this procedure, the number of pairs for comparison can easily grow. However, with the current computational power, even on a standard PC, the tool can do its job in a matter of a few hours.

The plagiarism detector can be used for all department laboratory work, irrespective of the programming language used, including assembly language and microprogramming sources. To perform its job, it preprocesses each input file set, finding any occurrence of a set of per-language predefined patterns known to be its primitive building blocks, such as assembler mnemonics, language reserved words, or standard library function names. In this way, it gets a string signature for each input file set. This signature does not pay attention to basic source changes, such as variable renaming or modified comments; neither does it take into account the current source layout. Instead, it focuses on the underlying logical structure of the program. Students who do not know how to do the exercise cannot easily change this structure. Working with this signature is enough to detect whether any program fragment has been copied, even if it has been moved around from its original position.

This tool does not provide certainty of plagiarism but a heuristically ordered list for manual examination for plagiarism. It cannot be considered as an automatic tool but should be viewed as a decision assistant because it does not tag plagiarism suspects. Moreover, because of the unpleasant nature and the relevance of this problem, the final decision must always be taken after careful eye inspection, and plagiarism must only be declared when it is viewed as evident by all the laboratory teachers. This plagiarism detection methodology has shown its validity because every published plagiarism case has been confirmed as positive by the students concerned. There may have been some false-negative cases, i.e., plagiarized works being considered as original. However, the authors were concerned with minimizing false-positive cases. Ultimately, publishing detected cases of plagiarism led students to think twice before trying to submit a copied work. Unfortunately, these data can be only indirectly derived from the experience gathered over several years in this department and cannot be compared with data for other institutions. This situation is a result of the absence of data published by other academic groups about plagiarism, mainly because this information is usually tagged as classified material.

IV. Implementation of the Automated Laboratory Work Environment

Different features of this laboratory work management environment have been successfully improved several times over the years in which it has been in use, ranging from the necessary elimination of some bugs to some increases in its functionality for both the student and the instructor. The last important change was the replacement of the previous specific and alphanumeric student interface by a Web-based interface in accordance with current trends. This Web-based interface is now standard for students.

This section describes some of the most important implementation aspects of the tools constituting the automated environment.

A. *Student File Delivering and Web Interface*

The set of objects on which the laboratory work automation environment is based has been illustrated in Fig. 1. An example of one of the Web-based interface pages is shown in Fig. 2. This interface provides the already logged-in student with a set of menu entries.

To avoid possible student–server communication interceptions, the login page requires a secure (HTTPS[3]) protocol.

The student interface has been developed and implemented based on open source tools to ensure high compatibility with almost any Unix computer. Specifically, the Web-based interface is supported by an Apache Web Server with the PHP[4] module enabled. It communicates with the laboratory work server through PHP scripts and TCP/IP[5] sockets. Students identify themselves by means of a login and password pair. The Apache Web Server sends this pair to the laboratory work server, where it is authenticated, allowing or denying entry to the system. To keep the connection state up to date, the PHP script uses the tools provided by the PHP core session manager. Because the Web session is composed of several TCP/IP connections, every connection to the laboratory work server is authenticated. Authentication is based on the user and password pair stored by the PHP Web session manager. The Web Server acts as a front end or proxy of the previous laboratory work server. This version has an alphanumeric user interface which is hidden by the proxy server.

The first time a student registers with the system, the correctness of his or her identification data is verified by searching a student's data file. Once verified, an entry is added to a registered students file, after which the server provides the client with a short-lived cookie, identifying the session. This cookie will be needed to correctly complete any sensitive transaction.

[3]Most of the acronyms used in this section are well known in the Web community: HTTPS is the secure version of Hypertext Transfer Protocol.

[4]PHP is a recursive acronym for PHP Hypertext Preprocessor.

[5]TCP/IP is the standard Transport Control Protocol/Internet Protocol.

When the cookie validity has expired, or after clicking the QUIT button, the server will be allowed to provide only generic information: news and help pages. Student files are uploaded by means of a POST command (multipart, file-type transfer) of the HTTPS protocol.

The most important reliability aspect deals with what to do when communication breaks down. To avoid file deposit errors, a conservative handshaking protocol is implemented; a transaction cannot produce any status modification until it has completely finished.

B. Automatic Project Evaluator

Automatic project evaluation is based on executing a set of tests for each laboratory exercise of every student. The laboratory work submitted by the student is executed for each test with an input data set previously established by the teacher. Output data generated by the test is compared with the correct set of results, already known by the evaluator. Some of the tests are executed as separate processes, directly running on a Unix computer. This behavior is observed in the operating system laboratory tests. However, computer architecture laboratory tests use an emulator because they are difficult to execute directly on a computer running in user mode. The evaluator is flexible enough to manage different programming languages and even different programming paradigms.

A test is usually executed in batch mode, which is done by building a program with the student laboratory work and the main function provided by the teacher. This execution is prepared in such a way that it will make calls to the student functions. For example, if the test is to be directly executed as a Unix process, the command for starting it, will be as follows:

```
Lab_work_name<input_data>results.
```

On the other hand, if using an emulator, the command to be executed will be as follows:

```
Emulator < input_data+student_code > results.
```

The automatic project evaluator is a tool developed using both scripting language and compiled C programs.

C. Plagiarism Detection Assistant

This tool works on the basis of the set of signatures found for each student input file set. The detection assistant is a C program that compares each signature with all the others, evaluating four different similarity-related criteria for each pair: the first criterion gives the size of the longer sequence of patterns common to both files, the second one calculates the cumulative value of the size of all the common sequences of patterns, the third gives a normalized value of the second as a percentage of its hypothetical maximum, and the fourth counts the number of patterns common to both files, irrespective of their relative positions, giving a normalized result. In other words, it gives the percentage of intersection between the histograms of their respective patterns.

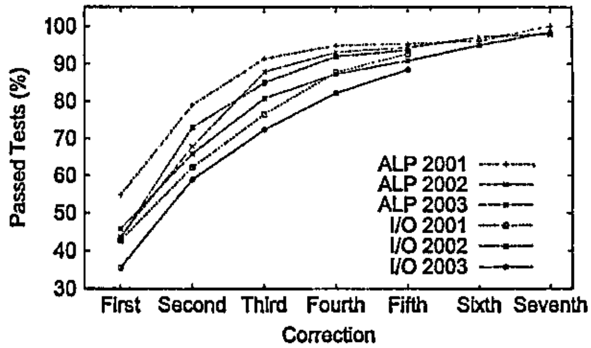


Fig. 4. Average evolution of passed tests.

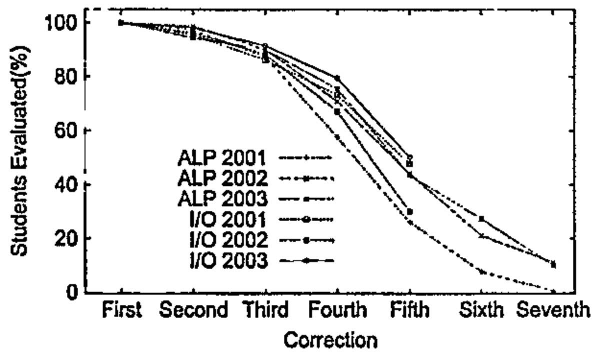


Fig. 5. Percentage of evaluated students at each correction.

The detector generates a file with a line for each compared pair with the four similarity measures. When this file is sorted by each criterion, it provides teachers with hints about the student file pairs more likely to contain similar code fragments. Depending on the number of students and the number of previous similar exercises in the repository, the total of compared pairs can easily reach a quarter of million. Usually, however, only the first 30 entries per criterion are used for inspection. Therefore, although these represent only an insignificant part of the total, they are enough to find positive cases and to decide when to stop. This explanation gives an idea of the soundness of this tool.

V. Results and Experiences

The benefits gained by students using these tools have been analyzed. On the basis of the experience acquired over the last three years, both in assembly language programming (ALP) and in I/O laboratory work, the authors have some measures asserting these benefits. Figs. 4 and 5, and Table I show a summary of the learning results achieved.

Table I shows the number of students used to compute Figs. 4 and 5. For example, in 2001, and considering the first correction of the I/O laboratory work, the evaluation was carried out for 381 students. In the second correction of the same laboratory work, 364 students were evaluated. These results mean that 17 out of 381 students used the evaluation tool just once, probably because they passed all of the tests.

Fig. 4 shows the average percentage of tests passed at each of the chances students have to use the automatic evaluator. At the first try, students pass on average under 50% of the tests, but

TABLE I
NUMBER OF SAMPLES USED FOR EVALUATION RESULTS

	I/O 2001	I/O 2002	I/O 2003	ALP 2001	ALP 2002	ALP 2003
First	381	362	355	377	274	304
Second	364	349	349	367	270	287
Third	328	320	325	330	247	274
Fourth	279	243	282	218	194	230
Fifth	182	109	179	99	121	133
Sixth	not available			29	58	83
Seventh	not available			2	31	31

finally, after the last try, the pass rate is over 95%.[6] Looking at the first three attempts, the authors find that the pass rate rises by 50% at the second correction and almost 100% at the third one. This evolution shows that the feedback provided by the automatic evaluation tool helps students to achieve the main laboratory work objective—to learn the principles of ALP or I/O techniques.

In this environment, students start the laboratory work with a previously established maximum number of allowed corrections or evaluation attempts. Each student can use from one to a maximum of such corrections. Fig. 5 shows the average percentage of students who use a given correction. Obviously, all students use their first correction, but less than 50% use the fifth one. This figure provides an indication of how difficult a specific laboratory exercise is. One has to take into account that a student can make use of the last correction and achieve the objective. About 85% of all students usually pass the laboratory work in the number of opportunities provided by the teacher.

The authors find that ALP laboratory exercises provide better results than I/O laboratory work, i.e., students pass more tests and do so earlier. There are several reasons for this result: I/O involves students acquiring more complex concepts, such as asynchronous events or even race conditions. Another reason, which is very specific to this course organization, stems from this first use of ALP programming in the first laboratory exercise in which students have to interact with an automatic management environment. Therefore, students have to be given a higher number of available corrections for this laboratory work than for the I/O exercise, where students are already familiar with the methodology to be used to complete the laboratory work.

The impact of the plagiarism detection tool can be analyzed with respect to the number of confirmed cases of plagiarism. Fig. 6 shows the evolution of plagiarism detected over the last three academic years in different laboratory exercises carried out in different courses. The laboratory work in question belongs to two different courses of operating systems, both of which deal with mini-shell programming (MSH) exercises. The third exercise is part of the computer structure laboratory course, and it handles ALP.

Notably, a lower percentage of plagiarism is usually detected in ALP (about 4% to 5%). Less detection of plagiarism is probably a result of the important specification changes made to this laboratory exercise in each new academic year. The value for the other laboratory exercises is usually under 10% of plagiarism. From previous experience in this environment, the authors consider this figure to be good, especially realizing that the work to be done by students is almost the same, or even exactly the same, every year.

[6]Not all the tests need to be passed to pass the course.

VI. CONCLUSION

An automatic management environment for laboratory work used at the Computer Engineering School of UPM has been presented. This environment is currently used to cover the practical part of teaching basic and advanced courses of computer architecture at the Department of Computer Systems Architecture and Technology, where there are more than 400 students and several laboratory exercises per year.

The only affordable way of managing these mass courses has been to use automation to assist instructors. There are other options, such as limiting the number of laboratory exercises, setting very simple exercises, or not checking result correctness; All of them lead to poor quality laboratory work and also reduce student learning options. Consequently, several tools have been developed to automate laboratory work management.

This environment has proved to be useful not only for mass courses but also for distance education. Students can interact with teachers, but they can also directly interact with the evaluation system through the delivery agent. This system is used for students to send their laboratory work files and get back the results. Students do not need to be at the school; they can work from elsewhere.

The automatic project evaluator exhaustively tests the correctness of each student laboratory exercise. The teacher provides a set of tests that try to cover the whole specification. Each failed test gives students enough feedback to correct their laboratory work and to learn from their mistakes.

Teaching experience has confirmed that there are always some students trying to pass the laboratory exercises without doing their work, by copying from other students. This plagiarism detector has been so effective that the authors can even conclude that it has dissuaded students from copying. Proof of this fact is that there are fewer students trying to deceive instructors each year.

With an automatic laboratory work management environment, the instructors have been able to maintain several high-quality projects per year, while being able to evaluate the correctness of the results of all students.

This experience shows that students achieve the learning objectives even when they are part of a mass course. This conclusion can be derived from Figs. 4 and 5, which illustrate the learning curves followed by students. At the beginning of the laboratory work, students usually pass only a few of the tests. However, these students who start their laboratory work early achieve better final results.[7] At the end, most of the students are able to pass all the tests. Passing assures that they have acquired the main concepts related to the laboratory work.

[7]To encourage students to meet the deadline, teachers allow them to use a *free* evaluation. The evaluator is configured so that if it is used early in the available period, students will not consume any of the limited number of attempts.

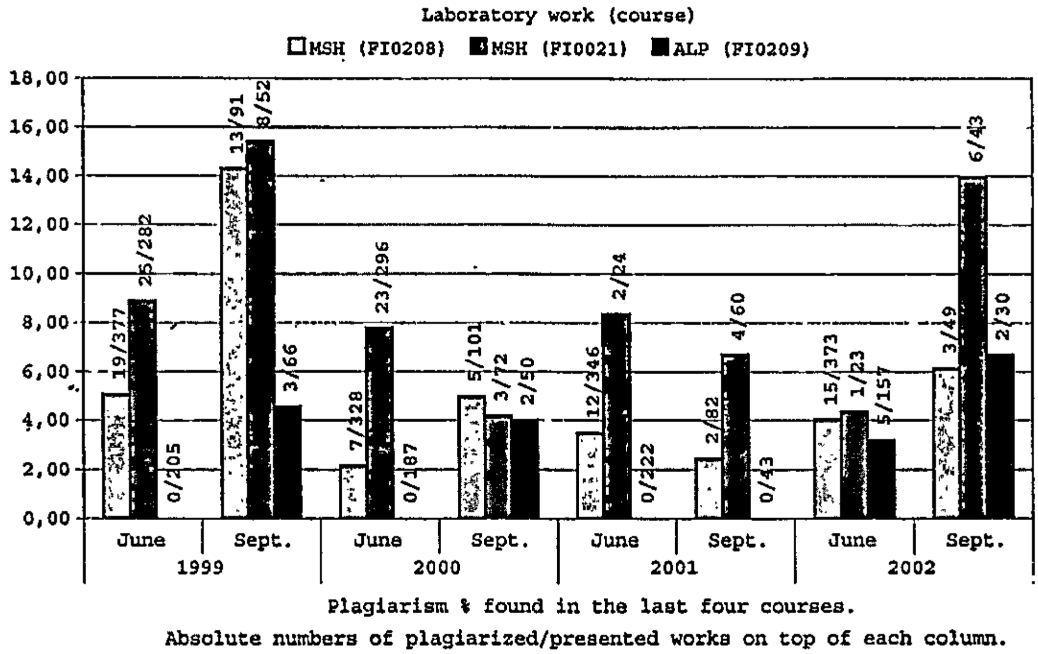


Fig. 6. Evolution of detected plagiarism.

This conclusion is also supported by the following data.

- ALP laboratory work (June 2002). The automatic evaluator worked with a set of 54 tests.
  - —90% of students starting the laboratory exercise early in the course passed all the tests.
  - —50% of students starting the laboratory work late passed all the tests. These students did not use the free evaluation available at the beginning of the laboratory work period.
  - —Jointly, 68% of students passed all the tests in the allotted time.
- I/O laboratory work (February 2002). The automatic evaluator worked with a set of 45 tests.
  - —79% of students starting the laboratory exercise early in the course passed all the tests.
  - —49% of students starting the laboratory work late passed all the tests.
  - —Jointly, 69% of students passed all the tests in the allotted time.

The free early evaluation is the only unusual action taken by teachers. No additional schedule is imposed on students who are allowed to use the limited number of previously established evaluations. This approach allows students to plan their work on the project as is done in real-world working practices.

In summary, this automated environment has proved to be very useful for managing mass courses and even distance education. Students using this environment are much more autonomous. They can use received feedback to think about their mistakes and to learn by themselves. The environment also makes it easier for instructors to be conscious of the importance of developing good specification documents. It will be better for students and also will prevent teachers from having to spend time answering unnecessary questions.

The realization that different laboratory exercises use the same automated environment proves its flexibility. Currently, it is used to cover the practical part of microprogramming, shell scripting, assembly language, I/O, and C programming. It can also be used to manage practical exercises for other programming languages, even object-oriented languages, such as Ada or C++.

By using the described environment, a larger number of students are receiving better technical education without having to increment the academic load of teachers.

References

[1] M. Castro *et al.*, "Examples of distance learning projects in the European community," *IEEE Trans. Educ.*, vol. 44, no. 4, pp. 406–411, Nov. 2001.

[2] H. Latchman, C. Salzmann, D. Gillet, and H. Bouzekri, "Information technology enhanced learning in distance and conventional education," *IEEE Trans. Educ.*, vol. 42, no. 4, pp. 247–254, Nov. 1999.

[3] A. Tartaglia and E. Tresso, "An automatic evaluation system for technical education at the university level," *IEEE Trans. Educ.*, vol. 45, no. 3, pp. 268–275, Aug. 2002.

[4] J. English and P. Sivitier, "Experience with an automatically assessed course," in *Proc. 5th Annu. SIGCSE/SIGCUE Conf. Innovation Technology Computer Science Education (ITiCSE 2000)*, Jul. 2000, pp. 168–171.

[5] L. Malmi, A. Korhonen, and R. Saikkonen, "Experiences in automatic assessment on mass courses and issues for designing virtual courses," in *Proc. 7th Annu. SIGCSE/SIGCUE Conf. Innovation Technology Computer Science Education (ITiCSE 2002)*, Jun. 2002, pp. 55–59.

[6] A. Korkohen *et al.*, "Does it make a difference if students exercise on the Web or in the classroom?," in *Proc. 7th Annu. SIGCSE/SIGCUE Conf. Innovation Technology Computer Science Education (ITiCSE 2002)*, Jun. 2002, pp. 121–124.

[7] R. Saikkonen, L. Malmi, and A. Korkohen, "Fully automatic assessment of programming examples," in *Proc. 6th Annu. SIGCSE/SIGCUE Conf. Innovation Technology Computer Science Education (ITiCSE 2001)*, Jun. 2001, pp. 133–136.

[8] K. Diefendorff and M. Allen, "Organization of the Motorola 88 110 superscalar RISC microprocessor," *IEEE Micro*, vol. 12, no. 2, pp. 40–63, Apr. 1992.

[9] S. Rodriguez, M. Garcia, and R. Mendez, "Teaching computer architecture with a new superscalar processor emulator," in *Proc. 4th Annu. SIGCSE/SIGCUE Conf. Innovation Technology Computer Science Education (ITiCSE 1999)*, Jun. 1999, pp. 99–102.

[10] B. Mott. (2004) Bsvc: A microprocessor simulation framework. [Online]. Available: http://www.redlinelabs.com/bsvc/

[11] M. Khalifa and R. Lam, "Web-based learning: Effects on learning process and outcome," *IEEE Trans. Educ.*, vol. 45, no. 4, pp. 350–356, Nov. 2002.

[12] A. Caruara, B. Ramaseshan, and M. Ewing, "The effect of anomie on academic dishonesty among university students," *Int. J. Educational Management*, vol. 14, no. 1, pp. 23–30, 2000.

Antonio García received the B.S. degree in computer engineering and the Ph.D. degree from the Technical University of Madrid, Madrid, Spain, in 1993 and 2001, respectively.

He is an Associate Professor in the Department of Computer Systems Architecture and Technology at the Technical University of Madrid, Madrid, Spain. His teaching interests include computer architecture, parallel and distributed systems, and hardware description languages.

Santiago Rodríguez received the B.S. degree in computer engineering and the Ph.D. degree from the Technical University of Madrid, Madrid, Spain, in 1990 and 1996, respectively.

He is an Associate Professor in the Department of Computer Systems Architecture and Technology at the Technical University of Madrid. His teaching interests include computer architecture and fault-tolerant computers.

Francisco Rosales received the B.S. degree in computer engineering from the Technical University of Madrid, Madrid, Spain, in 1991.

He is an Assistant Professor in the Department of Computer Systems Architecture and Technology at the Technical University of Madrid. His teaching interests include operating systems, simulation, and system-level programming.

José L. Pedraza received the B.S. degree in electronics and the Ph.D. degree from the University of Salamanca, Salamanca, Spain, in 1981 and 1987, respectively.

He is an Associate Professor in the Department of Computer System Architecture and Technology at the Technical University of Madrid, Madrid, Spain. His teaching interests include basic computer architecture, processor parallelism, and memory systems.

Envisioning Intelligent Information Technologies through the Prism of Web Intelligence

Users and businesses alike get to turn their raw data into new science, technology, and money.

BY NING ZHONG, JIMING LIU, AND YIYU YAO

Intelligent information technologies (iIT) focus on Web intelligence (WI), emphasizing the hybridization of techniques, along with multi-phase, distributed, and parallel processing. WI promises useful contributions to e-business intelligence and other Web-based technologies, including e-finance, e-science, e-learning, and e-service, and represents a significant benefit in IT development [9, 10, 12]. Although many IT techniques have been developed for intelligent information processing, even the most advanced are not yet mature enough to solve complex real-world problems. iIT can be regarded as the new generation of IT, encompassing the theories and applications of artificial intelligence (AI), pattern recognition, learning theory, data warehousing, data mining, knowledge discovery, grid computing, ubiquitous computing, autonomous agents, and multi-agent systems in the context of IT applications (such as e-commerce, business intelligence, social intelligence, knowledge grid, and knowledge community).

When investigating the future of IT, the computer science community should adopt an iIT perspective for several reasons. First, living in an information age, we are constantly developing new information media and technologies. Even personal information has become an important commodity. Enormous numbers of new data records are generated every second of every day. It must be summarized and synthesized to support problem solving and decision making in business, science, government, and university organizations. The continued growth of related data collection efforts ensures that the fundamental problem addressed by iIT—how one understands and uses one's data—will continue to be critical.

In the recent sea change computer scientists have experienced in AI research, it is now more common to build on existing theories than propose new ones, base claims on rigorous theorems or hard experimental evidence rather than on intuition, and be relevant to real-world rather than toy applications. iIT is an interdisciplinary field. Solving real-world problems involves techniques developed not only in the AI community but in other related communities, including statistics, cognitive science, and neuroscience. iIT also facilitates development of human and social intelligence.

Data is the source of human knowledge. By analyzing and using it, software engineers are able to do two things:

- Turn it into new science and technology, helping discover new scientific laws, protect and develop the environment in which we all live, and facilitate development of new science and technology; and
- Turn it into money, using it to make strategic and tactical business decisions, win new customers, retain existing customers, and reduce the cost (and waste) of doing business.

> The W4 generation will enable us to gain practical wisdom simply from living, working, and playing, in addition to conventional information search and knowledge queries.

A key iIT characteristic is the ability to combine AI, computational intelligence, WI, and intelligent agents in the design and implementation of intelligent Web-based information systems [10]. iIT development should thus be based on three main factors:

- New requirements for real-world applications (such as those in e-business, including e-commerce, e-finance, e-service, e-science, e-learning, e-government, and e-community);
- New architectures and methods for data repositories and data mining applications that satisfy these requirements; and
- New platforms and methods for large-scale distributed data sources that satisfy these requirements.

The key is being able to deal with the scalability and complexity of real-world problems. Doing so involves at least three major development methodologies: The first is hybridization, which optimally utilizes the advantages of existing methods (such as logic, including nonclassical logic, artificial neural networks, probabilistic and statistical reasoning, fuzzy sets, rough sets, and genetic algorithms).

Next is multi-phase process, or the methodology needed to solve complex real-world problems. Any future information system will have to integrate multiple subsystems. Processing information in them will not necessarily follow a well-defined linear sequence. Their operation involves multiple interrelated, perhaps iterative, phases. The dynamic organization of these phases is crucial to a system's viability.

The third is distributed and parallel processing. The Web's distributed, decentralized control means distributed and parallel processing is a must for the next generation of IT. Grid computing makes this a relatively easy task. The full potential of distributed and parallel processing must be realized in new-generation information processing systems.

In Data Mining

The three methodologies—hybridization, multi-phase process, and distributed and parallel processing—have been cited in many studies of intelligent agents, WI, and data mining. Their application is illustrated in the field of data mining. Data mining may be viewed as an interdisciplinary field combining results from many other fields. In order to systematically deal with real-world data, a useful methodology involves constructing a hybrid system for data mining-related processes (such as data selection, feature extraction and reduction, knowledge discovery, and visualization).

Data mining usually involves multiple steps, including data preparation, preprocessing, search for hypothesis generation, pattern formation, knowledge evaluation, representation, refinement, and management. It may also be iterative until the mined result is satisfactory for the user's purpose [2, 11]. As different types and sizes (gigabytes or even terabytes) of data are accumulated on multiple sites in large organizations, a particular user may need to access many data sources. The system must therefore support distributed mining, combining partial results into a meaningful whole.

How WI Represents iIT

The study of WI, introduced in [5, 9, 10, 12] explores the fundamental roles and practical effect of AI[1] (such as knowledge representation, planning, knowledge discovery and data mining, intelligent agents, and social network intelligence), as well as advanced IT (such as wireless networks, ubiquitous devices, social networks, and data/knowledge grids) on next-generation Web-based products, systems, services, and activities. On the one hand, WI applies results from existing disciplines to a totally new domain. On the other, WI introduces new problems and challenges to the established disciplines. WI may be viewed as an enhancement and/or extension of AI and IT.

Internet computing research and development in the next decade will be WI-centric, focusing on how to best use widely available Web connectivity. The new WI technologies will aim to satisfy five main post-industrial human needs [5]:

- Information empowerment;
- Knowledge sharing;
- Virtual social communities;
- Service enrichment; and
- Practical wisdom development.

One promising paradigm shift on the Web will be driven by the notion of wisdom, and developing the World Wide Wisdom Web (the Wisdom Web, or W4) will be a tangible goal for WI research [5]. The W4 generation will enable us to gain practical wisdom simply from living, working, and playing, in addition to conventional information search and knowledge queries.

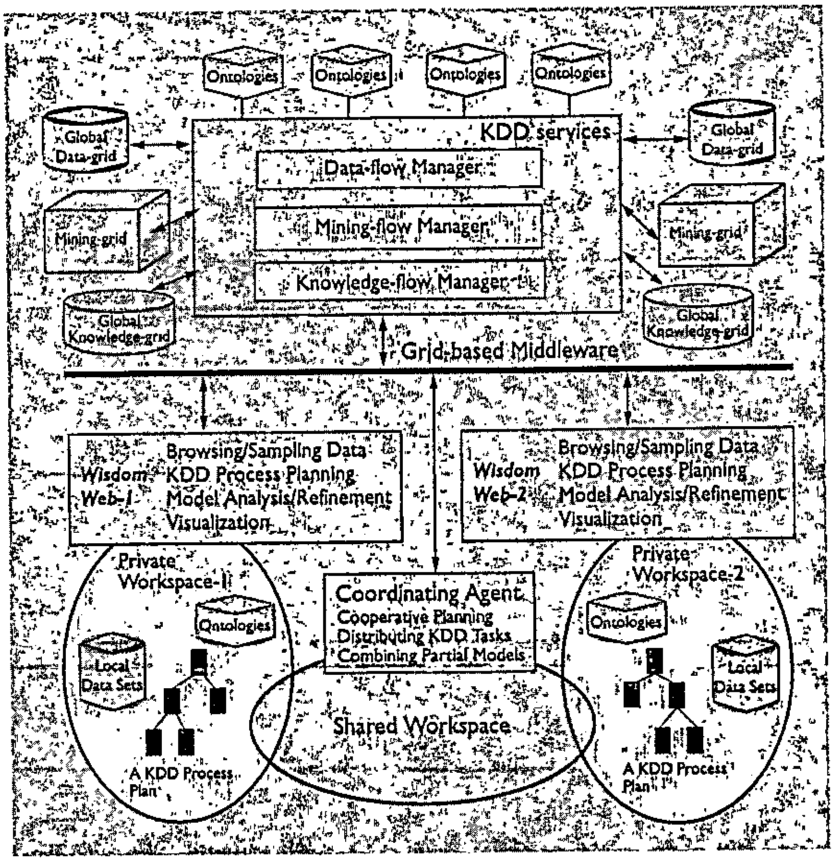
WI has great potential as a key iIT in intelligent enterprise portals for e-business intelligence, enabling organizations to create a virtual enterprise where production steps are outsourced to multiple partners. Many organizations today implement a corporate portal first, then grow it into more of an intelligent B2B portal. By using a portal to link back-end enterprise systems, an organization can manage the complex interactions of its virtual enterprise partners through all phases of the value and supply chain.

Developing intelligent enterprise portals involves a deep understanding of both centralized and distributed information structures on the Web. Information and knowledge on the Web are either globally distributed via the multilayer infrastructure of Web protocols or located locally, centralized on an intelligent portal providing Web services. However, neither approach is perfect. As pointed out in [1], the intelligent portal approach limits the uniformity of data formats and access, while the global semantic Web approach faces limitations involving combinational complexity.

Addressing these issues involves developing and using a Web-based problem-solving system for portal-centralized, query-answering intelligent Web services and decision making [9]. The core of such a system is the Problem Solver Markup Language (PSML)—designed to represent multiple information for Web-based problem solving—and PSML-based distributed Web inference engines [5]. When developing intelligent portals based on WI technologies, PSML must provide the following support functions:

- Expressive power and functional support for complex adaptive, distributed problem solving;
- Automatic reasoning on the Web by incorporating globally distributed content and metaknowledge automatically collected from the Semantic Web and from social networks with local databases and knowledge bases;

[1]Here the term AI includes all aspects of iIT.



Multi-database mining grid architecture on the Wisdom Web for an e-business portal.

- Representation and organization of multiple data/knowledge sources for distributed Web inference and reasoning;
- Combined reasoning methods; and
- Personalized models of user behavior, dynamically representing and managing it.

One way to begin to implement a PSML is to use a Prolog-like logic language with agent technologies. In our 2004 experiments we used a knowledge acquisition and utilization system (KAUS) to represent local information sources, as well as for inference and reasoning. KAUS is a knowledge management system involving databases and knowledge based on an extended first-order predicate logic and data model [6]. KAUS enables representation of knowledge and data in the first-order logic with data structure for inference and reasoning, as well as for transforming and managing knowledge and data.

Using an information-transformation approach helps developers combine the Web's dynamic, global information sources with local information sources in an enterprise portal for decision making and e-business intelligence.

Targeted telemarketing (also called direct marketing) is a new marketing strategy for e-business intelligence [8] that integrates Web-based direct marketing with other WI functions (such as Web mining and farming, the ontology-based search engine/question-answering system, personalized recommendation, and automatic email filtering and management) [9, 10]. Being able to track users' browsing behavior down to individual mouse clicks has brought vendors and their end customers closer than ever. It is now possible for vendors to personalize their product message for individual customers on a massive scale on the Web. Web farming extends Web mining into information analysis for Web-based information, including seeding, breeding, gathering, harvesting, and refining [4].

Customer data can be obtained from multiple customer touchpoints. In response, multiple data sources, including the Web, wireless communication and devices, call center, and brick-and-mortar store data, should be integrated into a single data warehouse to provide a view of customers, including their personal preferences, interests, and expectations. A multi-strategy, multi-agent data mining framework is required for the related analysis [11].

The main reason for developing a multi-agent data mining system is that various data mining agents must be able to cooperate in the multi-step data mining process, performing multi-aspect analysis, as well as multi-level conceptual abstraction and learning. Another reason is that a data mining task is decomposed into sub-tasks. They can be solved using one or more data mining agents distributed over different computers. The decomposition problem leads developers to the challenge of distributed cooperative system design.

A new infrastructure and platform is also needed as middleware to enable Web-based direct marketing for multi-aspect analysis from multiple data sources. One way to perform this direct marketing is to create a grid-based, organized society of data mining agents, or data mining grid, on the grid computing platform (such as the Globus toolkit) [3]. Various data mining agents are used for multi-aspect data analysis and targeted marketing tasks; they are organized into a grid with multi-layer components (such as data grid, mining grid, and knowledge grid) under the Open Grid Services Architecture, which responds to user queries by transforming them into data mining methodologies and discovering resources and information about them.

Computer scientists are able to use a conceptual model with three levels of dynamic workflows—data flow, mining flow, and knowledge flow—corresponding to the Grid with three layers—data grid, mining grid, and knowledge grid—in order to manage data mining agents for multi-aspect analysis in distributed, multiple data sources. The workflows are also useful for dynamically organizing status-based business processes, using ontologies to describe and integrate multi-data-source and grid-based data mining agents in data mining process planning [11]. They must also provide the following:

- A formal, explicit specification for the integrated use of multiple data sources in a semantic way;
- A conceptual representation of the types and properties of data and knowledge and data mining agents, as well as the relationships between data and knowledge and data mining agents;
- A vocabulary of terms and relationships to model the domain, specifying how to view the data sources and use data mining agents; and
- A common understanding of multiple data sources that can be communicated among grid-based data mining agents.

The figure here outlines an e-business portal's use of an agent-based multi-database mining grid architecture on the Wisdom Web. The system's architecture includes two main components: the multi-layer grid and the Wisdom Web. The multi-agent-based data mining grid architecture requires at least four types of meta agents:

Assistant. To help e-business users perform various work activities (such as browsing and sampling data and planning data mining process) and analyze/refine models on the Wisdom Web;
Interacting. To help users in their cooperative work activities (such as communication, negotiation, coordination, and mediation);
Mobile. To help move to global data grid services and execute within data grids; and
System. To administer multiple data mining agents to register and manage many components, monitor events and the status of the workspaces and agent meeting places, and collect relevant measurement following predefined metrics.

In such a multi-agent architecture, agents are created by and perform on behalf of users or other agents. They aim to achieve modest goals, reflecting the characteristics of autonomy, interaction, reactivity to environment, and proactive functions. The main components of the architecture in the figure interoperate in the following ways:

Establish workspaces. In large data mining processes, groups of people work as teams. Individual e-businesspeople have their own private workspace, while the group has a shared workspace. The people (or their agents) manage, control, and schedule work, accessing global databases and distributing data analysis tasks to the mining grid based on some resource allocation policy;
Create agents. Data mining agents support the e-business tasks and the data mining process; interacting agents support communications; mobile agents support data mining tasks within data grids; and system agents by default support the components in the architecture;
AgentMeetingPlaces. AMPs support communication among agent groups; some system agents (such as

WI has great potential as a key iIT in intelligent enterprise portals for e-business intelligence, enabling organizations to create a virtual enterprise where production steps are outsourced to multiple partners.

creation, deletion, and bookkeeping) are created by default to manage the AMPs;

Repositories. Repositories are global, local to one person, to a group of people, or distributed; local databases and model repositories are accessed from associated workspaces; global data grids are accessed only from controlling workspaces; and mobile agents travel to global data grids and execute there; and

Process models. Existing data mining process models are allowed into a workspace; for example, the planning and replanning techniques described in [11] can be applied.

The idea of e-business intelligence illustrates why it is so important for developers to study and use WI technologies systematically to deal with the scalability and complexity of real-world problems. Using WI technologies to intelligently manage, analyze, and use information from distributed data sources is a problem not only in e-business but in e-science, e-learning, e-government, and all WI systems and services. Developing enterprise portals and e-business intelligence is a good example of how software engineers might try to deliver such functions.

Developing the Wisdom Web represents a tangible goal for WI research [5, 10]. The paradigm of Wisdom Web-based computing aims to provide not only a medium for seamless information exchange and knowledge sharing but a type of artificial resource for sustainable knowledge creation, and scientific and social evolution. The Wisdom Web is likely to rely on grid-like service agencies that self-organize, learn, and evolve their actions in order to perform service tasks, as well as maintain their identities and relationships in communities. They will also cooperate and compete among themselves in order to optimize their own, as well as others', resources and utilities.

A notable research challenge in Wisdom Web-based computing is how to develop and demonstrate a ubiquitous agent community, that is, an intelligent infrastructure that enables agents to look ahead, then plan and deliver what users want [5]. It works like personal agency; for instance, it can help a user manage tedious daily routine activities (such as processing email, placing orders, organizing meetings, and downloading news).

Conclusion

iIT represents a paradigm shift in information processing, driven by WI, the Wisdom Web, grid computing, intelligent agents, autonomy-oriented computing, and other technological forces. WI is one of the most important of these forces, as well as a fast-growing iIT research field in its own right. iIT research could yield the new tools and infrastructure components necessary for creating intelligent portals throughout the Web. ◼

References

1. Alesso, H. and Smith, C. *The Intelligent Wireless Web.* Addison-Wesley, Reading, MA, 2002.
2. Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P., Eds. *Advances in Knowledge Discovery and Data Mining.* MIT Press, Cambridge, MA, 1996.
3. Foster, I. and Kesselman, C. *The Grid 2: Blueprint for a New Computing Infrastructure.* Morgan Kaufmann, San Francisco, 2004.
4. Hackathorn, R. *Web Farming for the Data Warehouse.* Morgan Kaufmann, San Francisco, 2000.
5. Liu, J. Web intelligence: What makes Wisdom Web? In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (Acapulco, Mexico, Aug. 9–15). Morgan Kaufmann, San Francisco, 2003, 1596–1601.
6. Ohsuga, S. Framework of knowledge-based systems: Multiple meta-level architecture for representing problems and problem-solving processes. *Knowledge Based Systems 3,* 4 (1990), 204–214.
7. Raghavan, P. Social networks: From the Web to the enterprise. *IEEE Internet Computing 6,* 1 (2002), 91–94.
8. Yao, Y., Zhong, N., Huang, J., Ou, C., and Liu, C. Using market value functions for targeted marketing data mining. *International Journal of Pattern Recognition and Artificial Intelligence 16,* 8 (2002), 1117–1131.
9. Zhong, N., Liu, J., and Yao, Y., Eds. *Web Intelligence.* Springer, 2003.
10. Zhong, N., Liu, J., and Yao, Y. In search of the Wisdom Web. *IEEE Computer 35,* 11 (2002), 27–31.
11. Zhong, N., Liu, C., and Ohsuga, S. Dynamically organizing KDD processes. *International Journal of Pattern Recognition and Artificial Intelligence 15,* 3 (2001), 451–473.
12. Zhong, N., Yao, Y., Liu, J., and Ohsuga, S., Eds. *Web Intelligence: Research and Development, LNAI 2198.* Springer, 2001.

Ning Zhong (zhong@maebashi-it.ac.jp) is head of the Knowledge Information Systems Laboratory and a professor in the Department of Systems and Information Engineering, Graduate School, Maebashi Institute of Technology, Japan. He is also an adjunct professor in the International WIC Institute at Beijing University of Technology, Beijing, China.

Jiming Liu (jiming@comp.hkbu.edu.hk) is a professor and head of the Department of Computer Science at Hong Kong Baptist University, Hong Kong, China. He is also an adjunct professor in the International WIC Institute at Beijing University of Technology, Beijing, China.

Yiyu Yao (yyao@cs.uregina.ca) is a professor in the Department of Computer Science at the University of Regina, Regina, Saskatchewan, Canada. He is also an adjunct professor in the International WIC Institute at Beijing University of Technology, Beijing, China.