Computers in Human Behavior 26 (2010) 825–834

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

# Assessing process in CSCL: An ontological approach ☆

Vive S. Kumar [a,*], Carmen L.Z. Gress [b], Allyson F. Hadwin [c], Phillip H. Winne [b]

[a] *Institute of Information Sciences and Technology, Massey University, New Zealand*
[b] *Faculty of Education, Simon Fraser University, Canada*
[c] *Faculty of Education, University of Victoria, Canada*

ARTICLE INFO

*Article history:*
Available online 14 September 2007

*Keywords:*
Computer-supported collaborative learning
Ontology
Trace data
gStudy

ABSTRACT

Educational technology innovations enable students to collaborate in online educational tasks, across individual, institutional, and national boundaries. However, online interactions across these boundaries are seldom transparent to each other. As a result, students are not motivated to share their best learning practices. Also, there is no singular basis on which one can compare learning practices of multiple students. In addressing these problems, we offer a solution that encourages students to record and share their learning interactions using our ontology-oriented theory-centric software tool. In doing so, students not only observe the products of their learning but also the process of how they learnt. These unique and computationally formal recordings of learning interactions not only allow educators to observe how learners learn, but also provide opportunities for learners to reflect on their understanding of meta-cognitive processes that they employed or neglected in their learning. Further, these recordings feed our software system to autonomously analyze students' learning behaviour and to actively promote self- and co-regulation among learners. This article presents the need for such a system, the architecture of the system, and concludes with key experimental observations from software prototypes.



## 1. Introduction

There is no singular basis upon which one can compare, in real-time, various learning practices of multiple students co-constructing knowledge and/or completing a collaborative project. Current analyses tools in CSCL, including self-report, content analysis, and observations (see Gress, Fior, Hadwin, & Winne, 2010 for more information), are informative yet do not meet the requirements necessary for this type of investigation. For instance, self-report questionnaires measure individual differences, which even if combined to form 'group differences' (a psychometrically dubious practice), they summarize learners' awareness, perceptions, recollections, and biases about learning processes (Winne & Jamieson-Noel, 2002). This information may or may not reflect actual learning practices. Content analysis of discussion and dialogues or talk-aloud protocols provide realms of invaluable information but because analysis is after the event, information gained does not provide opportunities for real-time feedback promoting learner evaluation, reflection, and adaptation. What this complex task requires are tools and methods that capture solo and shared learning activities. These tools would then categorize and summarize that information, according to principles based in evidence-based learning theories such as self-regulated learning (Winne, 2001; Winne & Hadwin, 1998; Zimmerman, 1989). The final component would be to provide that information to the learner(s) in the form of real-time feedback.

How does one go about designing a system that affords multiple collaborative opportunities while capturing all learner activities and the full context of their learning environments (e.g., open documents, notes being made, searches, conversations, and sharing) all on a standardized underlying metric upon which researchers can compare various learning practices of collaborating learners? One option is to build a system that traces each learner to learner and learner to computer interaction and design tools that utilize ontologies to analyze, categorize, and respond, all in real time. Trace data and ontological software tools provide a way to compare individual and group learning processes and products at a more fine-grained level than conventional measures of analysis. The use of ontologies as knowledge representations and knowledge sharing mechanisms is a century-old notion in philosophy. Over the past decade, researchers have extensively employed ontological knowledge representation in online learning environments. The ontological representation of the domain knowledge

☆ Support for this research was provided by grants from the Social Sciences and Humanities Research Council of Canada to P. H. Winne (410-2001-1263), and P. H. Winne (principal investigator), A. F. Hadwin and V. Kumar (co-investigators) (512-2003-1012), and from the Natural Sciences and Engineering Research Council of Canada to V. Kumar (www.lornet.org).

* Corresponding author.
*E-mail addresses:* V.S.Kumar@massey.ac.nz (V.S. Kumar), cgress@sfu.ca (C.L.Z. Gress).

0747-5632/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.chb.2007.07.004

and the interaction knowledge allows for a formal representation of concepts and the relationships between them, permitting researchers to share and view exact coding mechanisms and easy replication of these coding rules to new data.

The goals of this paper are many. First, we review the computer scientists' overture into education and collaborative learning, to acknowledge what other researchers have accomplished and what is yet to come. We start with a brief review of computing and learning, move to a discussion on CSCL software systems in general and the major accomplishments of the 1990s, then we discuss current objectives and highlight five core criteria we believe necessary for advancing CSCL systems. The remainder of this article presents our implementation of these five design criteria in a trace-based software system named *gStudy* (Winne et al., 2006), and its software extensions named MI-EDNA, SRLP, and MICE, demonstrate how the *system as a whole with its embedded ontologies* can enhance collaborative interactions and measurement.

## 2. Computing and collaboration – a review

### 2.1. Computing and learning

Research is ongoing in a number of frontiers to understand how learning infrastructure, cognitive tools, and social experiences sustain learning. Application of computing in the domain of learning, over the last two decades, resulted in a number of technological solutions. At the very beginning, Computer-Aided Instruction (CAI) brought the content of learning to an interactive electronic format – an electronic book. Student interactions in CAI systems were limited to mostly browsing the content. The arrival of Computer-Mediated Communication (CMC) techniques improvised human–computer interaction between the educational system and the learner. Successful CMC systems advocated that, unlike CAI, control of interaction be shared between the software system and the student. As one of the frontiers of online education, Intelligent Tutoring Systems (ITS) enable educational systems to deliver informed and pedagogically sound instructions. That is, the software system offers instructions in an autonomous fashion and is expected to be able to explain the reasoning behind its instructional plan. In a tangential direction, Courseware Management Systems (http://www.edutools.info/; http://en.wikipedia.org/wiki/List_of_content_management_systems; http://www.virtual-u.org/) offer techniques aimed at optimal online delivery and management of course material. In general, from a student's perspective, these systems facilitate him/her to engage in different dimensions of learning interactions.

Most of the CAI, CMC, CMS, and ITS systems offer one-to-one tutoring. In one-to-one computer-based instruction, the computer system interacts with one student and attempts to personalize the tutoring to the needs of this individual. With the underlying assumption that the computer application itself could be treated as an interacting partner to enliven and enrich the learning process, many of these systems attempted to capture student-system interactions assisting researchers in their development of models about how students collaborated with the software application. An example of one such student-system collaboration is found in Integration-Kid (Chan, 1991; Chan & Baskin, 1988). In Integration-Kid, a student collaborates with a system-simulated companion to learn the mathematical techniques of performing integration.

## 3. CSCL

A substantial leap in online education arrived in terms of Computer-Supported Collaborative Learning (CSCL). CSCL explores the social nature of learning focusing exclusively on enhancing mediated collaboration among multiple humans and humans and computer systems. In such one-to-many learning environments, the software system interacts with a select group of students who impart subject knowledge as a by-product of a collaborative learning strategy, which may be monitored and mediated passively or actively by the software system. Typically, collaborative learning systems are designed to concentrate on refining, integrating, and facilitating the learning process and content knowledge of students during collaborative activities.

The promise of collaborative learning environments is to provide an opportunity to facilitate learning in relatively realistic, cognitively motivating, and socially enriched learning contexts. For instance, a student might discuss the strategies to solve a problem in a problem-solving domain like trigonometry (trig tutor; Rajan, Patil, Anjaneyulu, & Srinivas, 1990) or practice the colloquial usage of a foreign language in a computer-aided language learning system (Kumar, Anjaneyulu, & Gupte, 1997) or offer offline contributions to a discussion group (Thomson, Greer, & Cooke, 1997). CSCL systems offer classroom-like environment with shared workspaces, on-line presentations, lecture notes, reference material, quizzes, student evaluation scores, and facilities for chat and online discussions (Soller, 2004). In these environments, students can discuss their learning strategies, their understanding, and their shortcomings with a group of fellow students, mediated by a software system. In doing so, they could take up roles to advise, to motivate, to criticize, to compete, or to direct their learning efforts towards better understanding of learning products and processes.

As Soller (2001) and Blumenfeld et al. (1996) point out, simply placing a student in a CSCL environment does not guarantee effective collaborative learning – there is every possibility of the student struggling to maintain a balance of participation, leadership, understanding, and encouragement. In an effort to alleviate this struggle, CSCL systems such as PHelps (Collins et al., 1997), I-Help (McCalla et al., 1997), and Helper's Assistant (Kumar, 2001) provide a facility for locating a peer helper who is ready, willing, and able to offer assistance and further support the peer helper. Many CSCL systems, discussed below, further address this very possibility.

As an early motivator of CSCL research, Miyake (1986) demonstrated that when compared to other types of learning, the bulk of constructive criticisms occurred while learning in collaboration – about 80% of self-critiquing (reflection) took place during collaborative learning compared to 20% in solo learning. Self-critiquing, which requires self-evaluation of the learner process and product to established standards, is a major contributor to learning effectiveness (Winne, 2001; Zimmerman, 2000). Miyake's results showed that learners might have missed the opportunity for better understanding if they had not collaborated. Blaye et al. (Blaye, 1989; Blaye, Light, Joiner, & Sheldon, 1990; Blaye, Light, Joiner, & Sheldon, 1991) showed that children who had previously worked as collaborative pairs on the task of planning and problem solving were twice as successful as children who had had the same amount of experience working alone. Translating such results from research in education into computing environments, Durfee, Lesser, and Corkill (1989) showed that computer software could be treated as a collaborating partner where the subject-level inconsistencies (a set of non-overlapping misconceptions) among collaborating peers were well-defined.

## 4. CSCL systems: the 1990s

A rather large number of CSCL systems were developed and experimented with in the 1990s. Here we highlight some of them in an effort to summarize how software systems were employed to mimic collaborative environments. ODISSEUS is an apprenticeship-