Running head: CHAT-BASED PROBLEM SOLVING IN SMALL GROUPS

Chat-based problem solving in small groups: Developing a multi-dimensional coding scheme

Jan-Willem Strijbos[1]

Leiden University

Gerry Stahl

Drexel University

[1] Correspondence can be sent to Jan-Willem Strijbos, Leiden University, Faculty of Social and Behavioural Sciences, Centre for the Study of Learning in Organisations, P. O. Box 9555, 2300 RB, Leiden, The Netherlands. E-mail: jwstrijbos@fsw.leidenuniv.nl

Abstract

Collaboration through chat communication has been primarily studied in dyadic settings in the domain of Computer-Supported Collaborative Learning (CSCL). In the 'Virtual Math Teams' (VMT) project small groups of three to five students collaborate through chat to solve a math problem. As part of a multi-method analysis toolkit a coding scheme was devised to conduct quantitative content analysis. During the calibration several methodological problems emerged. Firstly, the analysis required that the interaction structure (i.e., who responds to whom) would be reconstructed. Secondly, the diversity of processes of interest (e.g., conversational and problem solving acts) proved to be problematic. Although initially assumed independent, ties could not be avoided. Reliability computation for threading and coding proved challenging, for example not all utterances are valid analysis units for a dimension and results in overestimation of reliability. Reliability for most dimensions was satisfactory. Coding of the math dimension proved a bridge too far due to subtle nuances; hence methods like conversation analysis may be more applicable. The implications of the methodological issues for analysis of chat communication are discussed.

Keywords: content analysis, methodology, reliability, threading, coding

Problem solving through chat: Beyond dyadic interaction

Computer-supported collaborative learning (CSCL) is emergent in educational research. In the past decade debate has focused on theoretical, as well as to technical and pedagogical issues for CSCL. In comparison less attention has been paid to methodological aspects (Strijbos, Kirschner, & Martens, 2004a).

Collaborative learning research typically applies multiple methods to gather data, such as questionnaires and the (electronic) communication, each requiring a distinct analysis technique. For example multi-level modelling to investigate self-report questionnaires (Chiu & Khoo, 2003; Strijbos, Martens, Jochems, & Broers, 2004b) and content analysis to analyse the communication between the collaborating students (see e.g., Barron 2003; Gunawardena, Lowe, & Anderson, 1997; Veldhuis- Diermanse, 2002; Schellens & Valcke, 2005; Weinberger & Fischer, in press).

Although the number of studies reporting a specific methodology in more detail is rising – as well as the considerations for its application – many issues remain: especially with respect to the analysis of communication and written computer-mediated communication (CMC) in particular.

Early analyses methods of written CMC focused on surface characteristics, like the number of messages to determine participation degree (Harasim, 1993) and the mean number of words as a quality indication of a message (Benbunan-Fich & Hiltz, 1999). Later methods such as 'thread-length' (Hewitt, 2003) and 'social network analysis' (SNA; Lipponen, Rahikainen, Lallimo, & Hakkarainen, 2003) were added to the surface level repertoire. Presently the research community agrees that surface level methods provide an initial rough analysis, but more detailed analysis is needed to determine *why* one student contributes more or appears more influential.

Detailed analysis methods have flourished in the past decade. Roughly two approaches exist: on the one hand the quantitative approach where the frequencies of coded utterances are used for comparisons and/or statistical testing, and on the other hand the qualitative approach that applies

interpretative methods like case summaries (Lally & De Laat, 2003) and conversation analysis (Stahl, in press) to infer a trend or phenomenon (Miles & Huberman, 1994). Increasingly studies apply a mixed approach strategy to make sense of the data (for example Barron, 2003; Hmelo-Silver, 2003; Martinez, Dimitriadis, Rubia, Gómez, & De La Fuente, 2003, Strijbos, 2004).

Current CSCL research combines surface and detailed methods to unravel the communication and collaboration process (for example De Laat & Lally, 2004). Quantitative content analysis – in the remainder of this paper referred to as *coding* – for example, has been performed on a wide variety of written communication produced with specific tools: chat, forums and e-mail. One of the main components of coding methods is the chain of replies or sequence of interactions. The extent to which a specific sequence can be determined depends on the mode of communication: peer to peer, peer to group or many to many. Most forums use a predetermined threaded format that automatically inserts a response to a message as a subordinate object in a tree structure. In a similar vein a prefix is added to the subject header of an e-mail reply to indicate the sequence of the messages. In contrast to forums and e-mail communication, chat has no indicators identifying the chain of events. Due to its synchronous (same time, different place) character, chat is usually used as if one were communicating face-to-face. Most CSCL research using chat has focused on dyadic interaction (see the research on argumentation; Andriessen, Baker, & Suthers, 2003) and the communicative chain can be easily determined in this setting.

Yet, no attention has been awarded to the analysis implications of chat communication by a small group of three to five students with respect to the utterance sequence. Such small groups are the focus of the 'Virtual Math Teams' project (VMT) in which students discuss mathematics and solve problems through chat. We used small groups to decrease the impact of dropout, which was likely due to voluntary participation. In line with current multi-method strategies, the VMT project is developing an analytical toolkit combining qualitative (e.g., conversation analysis and

ethnography) and quantitative methods (e.g., threading/sequential analysis and coding). This paper discusses methodological issues that emerged during the development of a quantitative coding scheme. Specifically the threading of utterances ('who responds to whom'), as well as the coding of qualitatively different processes (like 'conversation' and 'problem solving') using the same data corpus, raised important methodological issues that have not been discussed in the literature thus far. Furthermore, reliability computation proved to be problematic as well.

The remainder of this article is structured as follows: first, we will describe the VMT project in more detail and more specifically the data-collection used. In the next section we will discuss our initial considerations with respect to the coding scheme, as well as the two methodological problems that emerged during the first three calibration trials: threading and coding of multiple processes. The next section elaborates on the issue of threading, the implications and solution(s), as well as the computation of inter-coder reliability. The following section addresses the issue of different processes – from hereon referred to as 'dimensions' – in detail, as well as inter-coder reliability. In the final section we will discuss the implications and recommendations for content analysis methodology of small group chat communication.

Introducing the Virtual Math Teams project

The Virtual Math Teams (VMT) project investigates small group collaborative problem solving in mathematics as a potential extension to The Math Forum's 'Problem of the Week (PoW)'. The project focuses on middle and high school students (grades 6 to 11). To investigate what kinds of support facilities are required and/or need to be designed – as well as apply existing and develop new analysis methods – an explorative data-collection was conducted. The students participated in one hour collaborative math problem solving sessions (powwow). They were spread across the United States and participation was voluntary. Students collaborated synchronously in groups of 3 to 5 students to solve math problems. AOL's Instant Messenger© software was used because

most students are already proficient users of this technology. A moderator invited each student to join a chat room and announced the start and closure of the session. The moderator did not take part in the problem solving discussion, s/he only answered technical and procedural questions (a specific protocol was designed to control moderator bias). Each powwow session is recorded as a transcript with the nickname of the student making an utterance, timestamp of the utterance, and the content of that utterance.

Analysing collaborative math problem solving through chat

Any content analysis starts with the development of a coding scheme. Most coding schemes that have been developed to analyse CSCL so far were guided by a specific research question. Some schemes focus on one construct, like the level of social knowledge construction (Gunawardena et al., 1997) or critical thinking (Newman, Webb, & Cochrane, 1995), whereas other schemes cover several different processes as part of the overall construct 'collaborative learning' and distinguish main and sub codes, like 'cognitive', 'affective', 'meta-cognitive processes' and 'rest' (Veldhuis-Diermanse, 2002), and 'coordinative', 'task-content', 'task-social', 'non-task' and 'non-codable' (Strijbos, et al., 2004b).

De Wever, Schellens, Valcke and Van Keer (in press) compared 15 coding schemes reported in the CSCL literature and they concluded that the reports differ greatly regarding the theoretical base of the coding scheme, as well as information about validity and reliability. In keeping with their plea to increase coherence between the theoretical base and the coding instrument, we will address our considerations in detail.

First of all, the unit of analysis needs to be determined. The granularity of the unit of analysis determines the accuracy of coding. This choice is affected by four contextual constraints: object of the study, nature of communication, collaboration setting and technological tool (see Strijbos, Martens, Prins, & Jochems, in press). In the VMT project we decided to use the chat line as the

unit of analysis mainly because it is the unit that the user defines. Furthermore, it allowed us to avoid all the issues of segmentation based on our (researcher) view and we empirically saw that chat users tended to only "do" one thing in a given chat line. It was also decided that the entire log would be coded including automatic entries by the chat system.

Secondly we determined what the coding scheme should capture. Obviously, we wanted to investigate the communicative process, problem solving and mathematical reasoning. An early version combined communicative statements with problem solving activities; however, this led to an enormous increase of sub codes. Subsequently, we decided to separate the communicative and problem solving coding and conceptualised these processes as independent dimensions, and a chat-line could have 0 or 1 code in any dimension. Our initial conceptualisation consisted of the conversational thread (C-thread; 'who replies to whom'), the conversation dimension (based on Beers, Boshuizen, Kirschner, & Gijselaers 2004; Fischer, Bruhn, Gräsel, & Mandl, 2002; Hmelo-Silver, 2003), the social dimension (based on Renninger & Farra, 2003; Strijbos et al., 2004b), the problem solving dimension (based on Jonassen & Kwon, 2001; Polya, 1985), the math move dimension (based on Sfard, 2002; Sfard & Linchevski, 1994; Sfard & McClain, 2003) and the support dimension (automatic system entries and utterances by the facilitator). The VMT coding scheme is illustrated in Appendix A: additions during calibration trials – see further sections – have been italicised for comparison.

Multi-dimensional coding schemes are not novel in CSCL research, but often not explicitly defined: Henri (1992), for example, describes five dimensions: participation, social, interactive, cognitive, and metacognitive; the social, cognitive and teaching presence instruments (Athabasca University) can be regarded as dimensions (see Anderson, Rourke, Garrison & Archer, 2000: Garrison, Anderson, & Archer, 2001; Rourke, Anderson, Garrison, & Archer, 1999). Fischer, Bruhn, Gräsel, and Mandl (2002) distinguish two dimensions: the 'content' and 'function' (in

terms of speech acts) of utterances, and Weinberger and Fischer (in press) use four dimensions: participation, epistemic, argument, and social. Yet, all these studies assign a single code to each utterance, or codes to multiple dimensions that differ in the unitization grain size (i.e., message, theme, utterance, sentence etc.). In contrast, the unitization in the VMT scheme is equal for all dimensions, therefore analyses can focus on one dimension but also combinations of different dimensions (e.g., 'offer + strategy', 'explain + strategy'); expanding the analytical scope.

After constructing the initial theoretical conceptualisation we first conducted several calibration sessions to determine whether the dimensions were capable to capture the behaviour of interest. These calibration sessions revealed two methodological problems: the chain of utterances – or in other words the 'threading' – and ties between dimensions assumed to be independent. The next section discusses the threading issue and followed by a section discussing the coding dimensions. Each section also addresses two reliability trials covering about 10% of the data (trial R1: 500 and trial R2: 450 lines) and reliability calculation issues that emerged.

Reliability of the VMT coding scheme

Reliability of threading

We started calibrating the conversation dimension and combined threading and conversational coding in a single analysis step, but quickly found out that threading should be assigned prior to the conversational codes. Whereas there is no confusion about the intended recipient in a dyadic setting (the other actor), students often communicate simultaneously in small groups making it easy to loose track of whom they should respond to. Coding the conversational dimension first requires the reconstruction of 'threading'. An example is provided in Table 1.

****

Insert Table 1 about here

****

Although the threading is performed separately from the conversational coding, it still requires the coder to be familiar with the codes to ensure that s/he knows which lines are considered for threading, because the conversational code depends on whether a thread is assigned. In addition, we introduced two codes to deal with fragmented statements ('setup' and 'extension') of a single author that span multiple chat lines. These fragments make sense only if considered together as a single statement. Thus, only one of the fragments is assigned a conversational code revealing the conversational action of the whole statement, and the remaining fragments are tied to that special fragment by using 'setup' and 'extension' codes. For example line 155 is an extension to 154 and together they form a 'request' and line 156 is a setup to line 158 forming a 'regulation'.

Delay proved to be important when assigning threading, for example lines 157 and 158 fully overlap (no delay) and the delay between lines 166 and 167 of 16 seconds reveals that the short utterance of 167 is more likely linked to 166 than 164, our reasoning is that it takes only a few seconds to type and submit this utterance and if line 167 was intended as a response to line 164 this utterance would have appeared before or simultaneous with line 166.

The calculation of 'threading reconstruction' reliability proved complicated as two coders can assign a threading indicator to a chat-line or not, to the same chat-line, or to a different chat-line. As a result it only a proportion agreement can be computed. We used three coders and computed three proportion agreement indices for each dyad:

- for the assignment of a thread by both coders;
- for the assignment of the same thread by both coders;

- for the assignment of different threads whenever both assigned a thread.

Table 2 present the results for both reliability trials for each pair of coders. The first trial (R1) consisted of 500 chat-lines and the second trial (R2) consisted of 450 chat-lines.

****

Insert Table 2 about here

****

In a similar vein the calibration session for the problem solving dimension revealed the necessity of a 'problem solving thread' to follow the co-construction of ideas and flow of problem solving acts (e.g., proposing a strategy or performing a solution step). Table 3 present the results for both reliability trials for each pair of coders.

It should be noted that the problem solving thread is often the same as the conversation thread and therefore the reliability indices are automatically slightly higher. In addition, the selection in R2 contained less problem solving utterances compared to R1, hence the problem solving thread is more similar to the conversational thread compared to R1 and therefore reliability is higher.

****

Insert Table 3 about here

****

A threshold for the proportion agreement reliability of segmentation does not exist in CSCL research nor in the domain of content analysis (see De Wever et al., in press; Neuendorf, 2002; Riffe, Lacy, & Fico, 1998; Rourke, Anderson, Garrison, & Archer, 2001). Various perspectives on the criterion value can be found in the literature. Given all perspectives a range of .70 to .80 for proportion agreement applies best. Combined results reveal for the conversational thread that on average in 80.7% of all instances both coders assign a thread. In 65.7% of all assignments by either coder is the same, and when both coders assign a thread 27.6% is different. The combined

results show that the reliability of conversational threading is quite stable. The degree to which conversational threading can be detected fits in the .70 to .80 range. The degree of same thread assignments is below this standard.

The results of both reliability trials reveal for the problem solving that on average in 84% of the all instances both coders assign a thread. Of all threading assignments by either coder 80.3% is the same, and when both coders assign a thread 8% is different. The degree, to which problem solving threading can be detected, as well as same thread assignments, fits in the .70 to .80 range. However, the results show that the reliability of the problem solving threading depends on the extent to which utterances actually contain problem solving content, so reliability will fluctuate between transcripts. Therefore the first trail can be a satisfactory lower bound: 77.1% for thread assignment and 71.2% for same thread assignment.

Reliability of coding

Given the impact of the conversational and problem solving thread – as well as the problem that ties between coding dimensions could not be avoided (e.g., offer and elaborate had to be linked to problem solving and math content). Throughout the calibration session's codes were added or changed, definitions adjusted, prototypical examples added, and rules to handle exceptions were established. In all, nine calibration trials were conducted prior to the two reliability trials. For the reliability trials we adopted a stratified approach: coders individually assigned the conversation threads, followed by a discussion to construct an agreed conversational thread after which each coder independently codes the social and support dimension. Next coders individually assigned the problem solving thread and followed by a discussion to construct an agreed problem solving thread after which each coder assigns problem solving and math move codes.

Appendix A shows the final coding scheme that was used for both reliability trials (for a more detailed description visit http://mathforum.org/wiki/VMT?ThreadAnalResults). The math move

dimension proved to be most difficult, because of many subtle nuances involved. We are able to detect mathematical content. Although a more specific typology is under construction other analysis methods like conversation analysis may be more applicable in the end to uncover this process in line with our preferred level of detail. Hence, we will not report a statistic for this dimension for either reliability trial.

Between both reliability trials several minor changes were made in the wording of a definition or adjusting a rule. A major change was made in the problem solving dimension where we added two codes after the first reliability trial. The code ‘corroborate/counter’ was added to highlight a connection with the conversational codes ‘agreement’, ‘disagreement’ or ‘follow’ with a specific reference to the problem solving, as well as to ambiguity in the interpretation of what the ‘check’ code in the problem solving dimension. Similarly, ‘clarify’ was added to highlight questions that asked for clarification with respect to problem solving. In addition, the definitions for the ‘check’ and ‘reflect’ code were adjusted, i.e. check was re-defined as referring to a solution step or result and reflect was re-defined as referring to level of conceptual/abstract reasoning.

Calculating the reliability for the conversation, social, support and problem solving dimension proved to be less straightforward than expected. Each chat line receives a conversation code, but not all chat lines are eligible to receive for example a problem solving code. The social, support and problem solving dimension only apply to a part of all chat lines. Simply adding a ‘no code’ category for all chat lines that were not coded by both coders results in a gross overestimation, yet only the chat lines that received a code by either coder at the valid pool of utterances ignores the ambiguous utterances that both coders considered but not coded.

Most coding schemes – whether they focus on single construct or different processes as part of the overall construct ‘collaborative learning’ – cover all units and/or the pool for specific parts can be easily determined, whereas the pool of valid units in the case of dimensions fluctuates (in

Figure 1 this fluctuation is depicted).

****

Insert Figure 1 about here

****

To determine the degree of reliability for each dimension we decided to calculate the number of agreed codes relative to the number of lines that were assigned a code, as well as the number of agreed codes relative to all chat-lines to determine the degree of overestimation. We used three coders. As suggested by De Wever et al. (in press) we calculated proportion agreement as well as Cohen's kappa to correct for chance agreement for each dimension, pair of coders, and for unit pools (not applicable for the conversation dimension).

In case where the pool of units consists of all units coded by either coder we need to insert these units in the kappa calculation. As Cohen's kappa does not consider the magnitude of the misses, all instances where only one coder of a pair coded a unit can be inserted as deviations off the diagonal in the kappa matrix. In the case where all units form a valid pool one code (no code) is added to the dimension, extending the kappa matrix with an extra coding category. Results for the conversation and social dimension are shown in Table 4 (proportion agreement = %, Cohen's kappa = K, and valid pool of units = U).

****

Insert Table 4 about here

****

We applied the .70 to .80 range for the proportion agreement. Again various perspectives on the criterion value for Cohen's kappa can be found in the literature. Given all perspectives we think the following criteria – intermediate strict and lenient – apply best: below .45 'poor', .45 to .59 'fair', .60 to .74 'good', and . 75 and above 'excellent' agreement beyond chance (see De Wever

et al., in press; Beers, Boshuizen, Kirschner, Gijselaers, 2005; Landis & Koch, 1977; Neuendorf, 2002). Mastery of the coding procedure is laborious. It takes about twenty hours of training and discussion with an experienced coder per dimension.

Proportion agreement for the conversation dimension is on average .695 (R1) and .727 (R2) and considered satisfactory. Cohen's kappa is on average .656 (R1) and .693 (R2), considered 'good'. Proportion agreement for the social dimension for R1 in the case of a relative unit pool is on average .524 and kappa is .419 (poor) and for R2 proportion is .565 and kappa .472 (fair). The units fluctuate on average 20 units for R1 and 12 for R2. When all units are used the proportion for R1 is .808 and kappa .627 (good) and for R2 .837 and kappa .686 (good). Overestimation is for R1 and R2 combined on average .278 for proportion agreement and .211 for kappa.

****

Insert Table 5 about here

****

Table 5 shows the results for the support and problem solving dimension .Proportion agreement for the support dimension for R1 in the case of a relative unit pool is on average .681 and kappa is .644 (good) and for R2 proportion is .784 and kappa .698 (good). The unit pool fluctuates on average 4 units for R1 and 2 units for R2. When all units are used proportion is for R1 .982 and kappa .805 (excellent) and for R2 .978 and kappa .865 (excellent). The average overestimation for R1 and R2 combined is .248 for proportion agreement and .164 for kappa.

Proportion agreement for the problem solving dimension for R1 in the case of a relative unit pool is on average .419 and kappa is .316 (poor) and for R2 proportion is .588 and kappa .519 (fair). The units fluctuate on average 19 units for R1 and 12 units for R2. When all units are used for R1 the proportion is .812 and kappa .584 (fair) and for R2 .827 and kappa .709 (good). The combined overestimation (R1 and R2) is on average of .316 for proportion and .229 for kappa.

Conclusion

Reconstructing the conversational and problem solving thread proved to be crucial. Overall, the coders are satisfactorily able to detect thread assignment (.80 for the conversational thread and .83 for the problem solving thread), and assign to a substantial degree same threads (.66 for the conversational thread and .79 for the problem solving thread).

Overall the reliability for the other dimensions – pool of units for social, support and problem solving consisting of all chat lines that are coded by either coder – ranged from 'poor' to 'good' in the first reliability trial with average proportion agreement of .419 to .695 and Cohen's kappa of .316 to .656. In the second trial they ranged 'fair' to 'excellent' with proportion agreement of .565 to .784 and Cohen's kappa of .472 to .698. Coding reliability increased for all dimensions in the second trail, although the increase for the social dimension was small.

For several dimensions the valid pool of units that can be considered for coding fluctuates. If we include all non-coded units by both coders in the calculation the reliability is overestimated to a degree of .248 to .316 (proportion agreement) and .164 to .229 (Cohen's kappa). Thus, a slight underestimation of 'valid' non-coded units is favoured given the substantial overestimation if all non-coded units are included.

Discussion

The aim of the research presented in this article was to develop a quantitative content analysis procedure for chat-based mathematical problem solving. During the course of the development, several methodological issues emerged that are not addressed in the CSCL literature, but which have important implications for content analysis methodology and practice.

To date CSCL studies using chat have focused on dyadic interaction. The Virtual Math Teams (VMT) project investigates small groups of 3 to 5 students solving math problems through chat. A coding scheme was devised to capture the processes of interest (conversation, social, support,

problem solving, mathematical reasoning, and threading) and during the first calibration trails it became apparent that the threading – 'who responds to whom' – had to be reconstructed prior to coding of the conversation, social and support dimension. Similarly, a problem solving thread was required prior to coding of problem solving and math moves. In essence the threading is a deep interpretation of what is going on the chat. Aggregating all coding divergence will result in very low reliabilities, thus agreement on threading prior to coding is required. We conducted two reliability trails and conversational and problem threading proved to be sufficiently reliable.

It can be argued that current chat technology does not accommodate small groups; hence the threading would be reduced to a pseudo-problem. Nevertheless, educational practice often uses a chat tool for small group discussions. In fact, turn-taking mechanisms have been implemented in several chat-tool used for research purposes, an example in dyadic interaction support is the TC3 environment (Van der Puil, Andriessen, & Kanselaar, 2004). Others researchers have developed 'threaded chat' (Smith, Cadiz, & Burkhalter, 2000) and explicit referencing (see ConcertChat©; Mühlpfordt & Wessner, 2005). Threaded chat appears very useful to assist both the students (and researcher) in making sense of the temporal order in which the chat lines are intended to be read by other students, or to reveal parallel communication strands (for an example of parallel strands in a VMT chat see Cakir, Xhafa, Zhou, & Stahl, 2005). Another promising innovation is a queue in Mediated Chat© version 3.0 (Pimentel, Fuks, & De Lucena, 2005) designed to prevent that messages are posted simultaneously. We are currently using ConcertChat© during the second data-collection wave.

Although usability tests indicate that threaded chat/referencing and a message queue appear to decrease ambiguity and confusion, both potentially increase self-censure. Students with a high status or a tendency to reply immediately regardless of the threading or queue functionality, may (unwillingly) hamper students with a lower status, slower writing pace and/or less confidence in

their own ability. They may decide to delete a contribution in favour of one that appears quickly, regardless of the problem solving progress potential. In sum, threaded chat and queuing appear to be promising, but systematic controlled studies are needed to determine their impact on collaboration.

With respect to the coding of the collaborative math problem solving chat the diversity of the processes of interest – conversational, social, support, problem solving and math moves – proved to be problematic. Although initially assumed independent, ties could not be avoided. Reliability computation for the social, support and problem solving dimension appeared problematic, e.g. not all chat-lines are valid analysis units for these dimensions and this leads to overestimation of the reliability. The degree of overestimation using all chat lines exceeds the underestimation for the case when only those chat lines coded by either coder are included in the calculation, a slight underestimation of 'valid' non-coded units is favoured given the substantial overestimation if all non-coded units are included. In the latter case the pool of units will fluctuate between trials and coder pairs, hence the valid pool of units should be reported (see for example Hurme & Järvelä, 2005, p. 6). Overall coding reliability increased for all dimensions during the second trail (but the increase for the social dimension was small): proportion agreement was satisfactory and Cohen's kappa ranged from 'fair' to 'excellent'. Irrespective of the reliability the coding scheme is a long-term effort and other dimensions are explored (e.g., problem solving progress).The mathematical dimension proved to be most difficult because of the subtle nuances involved: analysis methods like conversation analysis may be more applicable (Zemel & Xhafa, 2005).

CSCL is a young paradigm in educational research, but the increase of articles focusing on for example content analysis methodology signifies that the field is maturing. Hopefully, this article can serve as a developmental scaffold.

References

Anderson, T., Rourke, L., Garrison, D. R., & Archer, W. (2001). Assessing teaching presence in a computer conference context. *Journal of Asynchronous Learning Networks*, *5*(2). Retrieved August 8, 2005, from http://www.sloan-c.org/publications/jaln/v5n2/pdf/v5n2_anderson.pdf

Andriessen, J., Baker, M., & Suthers, D. (Eds.) (2003). *Arguing to learn: Confronting cognitions in computer-supported collaborative learning*. Dordrecht: Kluwer Academic/ Springer Verlag.

Benbunan-Fich, R., & Hiltz, S. R. (1999). Impacts of asynchronous learning networks on individual and group problem solving: A field experiment. *Group Decision and Negotiation*, *8*, 409-426.

Barron, B. (2003). When smart groups fail. *The Journal of the Learning Sciences*, *12*, 307-359.

Beers, P. J., Boshuizen, H. P. A., Kirschner, P. A., & Gijselaers, W. (in press). Computer support for knowledge construction in collaborative learning environments. *Computers in Human Behavior*.

Beers, P. J., Boshuizen, H. P. A., Kirschner, P. A., & Gijselaers, W. (2005, August). The analysis of negotiation of common ground in CSCL. In J. W. Strijbos & F. Fischer (Chairs), *Measurement challenges for collaborative learning research*. Paper presented in a symposium conducted at the 11th biennial EARLI conference, Nicosia, Cyprus.

Cakir, M., Xhafa, F., Zhou, N., & Stahl, G. (2005, July). *Thread-based analysis of patterns of collaborative interaction in chat*. Paper presented at the 12th international conference on artificial intelligence in education, Amsterdam, The Netherlands.

Chiu, M. M., & Khoo, L. (2003). Rudeness and status effects during group problem solving: Do they bias evaluations and reduce the likelihood of correct solutions? *Journal of Educational Psychology*, *95*, 506-523.

De Laat, M., & Lally, V. (2004). It's not so easy: Researching the complexity of emergent participant roles and awareness in asynchronous networked learning discussions. *Journal of Computer Assisted Learning, 20,* 165-171.

De Wever, B., Schellens, T., Valcke, M., & Van Keer, H. (in press). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education*.

Fischer, F., Bruhn, J., Gräsel, C., & Mandl, H. (2002). Fostering collaborative knowledge construction with visualization tools. *Learning and Instruction, 12,* 213-232.

Garrison, D. R., Anderson, T., & Archer, W. (2001). Critical thinking, cognitive presence, and computer conferencing in distance education. *American Journal of Distance Education, 15,* 7-23. Retrieved August 8, 2005, from http://communitiesofinquiry.com/documents/CogPresPaper_June30_.pdf

Gunawardena, C. N., Lowe, C. A., & Anderson, T. (1997). Analysis of a global online debate and the development of an interaction analysis model for examining social construction of knowledge in computer conferencing. *Journal of Educational Computing Research, 17,* 397-431.

Harasim, L. (1993). Collaborating in cyberspace: Using computer conferences as a group learning environment. *Interactive Learning Environments, 3,* 119-130.

Henri, F. (1992). Computer conferencing and content analysis. In A. Kaye (Ed.), *Collaborative learning through computer conferencing: The Najaden papers* (pp. 117-136). London: Spinger Verlag.

Hewitt, J. (2003). How habitual online practices affect the development of asynchronous discussion threads. *Journal of Educational Computing Research, 28,* 31-45.

Hmelo-Silver, C. E. (2003). Analyzing collaborative knowledge construction: Multiple methods for integrated understanding. *Computers & Education, 41,* 397-420.

Hurme, T.-R., Järvelä, S. (2005). Students' activity in computer-supported collaborative problem solving in mathematics. *International Journal of Computers for Mathematical Learning, 10,* 49-73.

Jonassen, D. H., & Kwon, H. I. (2001). Communication patterns in computer mediated and face-to-face group problem solving. *Educational Technology Research & Development, 49,* 35-51.

Lally, V., & De Laat, M. (2003). A quartet in E. In B. Wasson, S. Ludvigsen & U. Hoppe (Eds.), *Designing for change in networked learning environments* (pp. 47-56). Dordrecht: Kluwer Academic/Springer Verlag.

Landis, J., & Koch, G. (1977). The measurement of observer agreement for categorical data. *Biometrics, 33,* 159-174.

Lipponen, L., Rahikainen, M., Lallimo, J., & Hakkarainen, K. (2003). Patterns of participation and discourse in elementary students' computer-supported collaborative learning. *Learning & Instruction, 13,* 487-509.

Martinez, A., Dimitriadis, Y., Rubia, B., Gómez, E., & De La Fuente, P. (2003). Combining qualitative evaluation and social network analysis for the study of classroom social interaction. *Computers & Education, 41,* 353-368.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook.* London: Sage.

Mühlpfordt, M., & Wessner, M. (2005). Explicit referencing in chat supports collaborative learning. In T. Koschmann, D. Suthers & T. W. Chan (Eds.), *Computer supported collaborative learning 2005: The next 10 years!* (pp. 460-469). Mahwah, NJ: Lawrence Erlbaum Associates.

Newman, D. R., Webb, B., & Cochrane, C. (1995). *A content analysis method to measure critical thinking in face-to-face and computer supported group learning*. Retrieved August 11, 2005, from http://www.qub.ac.uk/mgt/papers/methods/contpap.html

Neuendorf, K. A. (2002). *The content analysis guidebook*. Thousand Oaks, CA: Sage publications.

Polya, G. (1985). *How to solve it*. Princeton: Princeton University Press.

Pimentel, M., Fuks, H., & De Lucena, C. J. P. (2005). Mediated chat development process: Avoiding chat confusion on educational debates. In T. Koschmann, D. Suthers & T. W. Chan (Eds.), *Computer supported collaborative learning 2005: The next 10 years!* (pp. 499-503). Mahwah, NJ: Lawrence Erlbaum Associates.

Renninger, K. A., & Farra, L. (2003). Mentor-participant exchange in the Ask Dr. Math service: Design and implementation considerations. In M. Mardis (Ed.), *Digital libraries as complement to K-12 teaching and learning* (pp. 159-173). ERIC Monograph Series.

Riffe, D., Lacy, S., & Fico, F. G. (1998). *Analyzing media messages: Using quantitative content analysis in research*. Mahwah, NJ: Lawrence Erlbaum Associates.

Rourke, L., Anderson, T., Garrison, D. R., & Archer, W. (2001). Methodological issues in the content analysis of computer conference transcripts. *International Journal of Artificial Intelligence in Education*, *12*, 8-22.

Rourke, L., Anderson, T., Garrison, D. R., & Archer, W. (1999). Assessing social presence in asynchronous text-based computer conferencing. *Journal of Distance Education*, *14*(3), 51–70. Retrieved August 8, 2005, from http://cade.athabascau.ca/vol14.2/rourke_et_al.html

Schellens, T., & Valcke, M. (2005). Collaborative learning in asynchronous discussion groups: What about the impact on cognitive processing? *Computers in Human Behavior*, *21*, 957-975.

Sfard, A. (2002). There is more to discourse than meets the ears: Looking at thinking as communicating to learn more about mathematical learning. In C. Kieran, E. Forman & A. Sfard (Eds.), *Learning discourse: Discursive approaches to research in mathematics education* (pp. 13-57). Dordrecht: Kluwer.

Sfard, A., & Linchevski, L. (1994). The gains and the pitfalls of reification: The case of algebra. *Educational Studies in Mathematics*, *26*, 191-228.

Sfard, A., & McClain, K. (2003). Analyzing tools: Perspectives on the role of designed artifacts in mathematics learning: Special issue. *Journal of the Learning Sciences*, *11*(2 & 3).

Smith, M., Cadiz., J. J., & Burkhalter, B. (2000). Conversation trees and threaded chat. In *Proceedings of CSCW 2000* (pp. 97-105). New York: ACM Press.

Stahl, G. (in press). *Group cognition: Computer support for building collaborative knowledge*. Cambridge, MA: MIT Press.

Strijbos, J. W. (2004). *The effect of roles on computer-supported collaborative learning*. Unpublished doctoral dissertation, Heerlen, The Netherlands, Open University of The Netherlands.

Strijbos, J. W., Kirschner, P. A., & Martens, R. L. (2004a). (Eds.) *What we know about CSCL: And implementing it in higher education*. Boston, MA: Kluwer Academic/Springer Verlag.

Strijbos, J. W., Martens, R. L., Jochems, W. M. G., & Broers, N. J. (2004b). The effect of functional roles on perceived group efficiency: Using multilevel modeling and content analysis to investigate computer-supported collaboration in small groups. *Small Group Research*, *35*, 195-229.

Strijbos, J. W., Martens, R. L., Prins, F. J., & Jochems, W. M. G. (in press). Content analysis: What are they talking about ? *Computers & Education*.

Van der Puil, C., Andriessen, J., & Kanselaar, G. (2004). Exploring relational regulation in computer-mediated (collaborative) learning interaction: A developmental perspective. *Cyberpsychology & Behavior*, *7*, 183-195.

Veldhuis-Diermanse, E. A. (2002). *CSCLearning? Participation, learning activities and knowledge construction in computer-supported collaborative learning in higher education*. Unpublished doctoral dissertation, Wageningen, The Netherlands, Wageningen University.

Weinberger, A., & Fischer, F. (in press). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*.

Zemel, A., & Xhafa, F. (2005, August). What's in the mix: Combining coding and conversation analysis to investigate chat-based problem solving. In J. W. Strijbos & F. Fischer (Chairs), *Measurement challenges for collaborative learning research*. Paper presented in a symposium conducted at the 11th biennial EARLI conference, Nicosia, Cyprus.

Appendix A: VMT coding scheme (dimension and codes that were added during calibration trials are italicised).

C-thread	**Conversation**	**Social**	**Support**	*PS-thread*	**Problem Solving**	**Math move**
Reply to U_i	No code	Identity self	Entry	*Connect to U_i*	Orientation	Counting
	State	Identity other	Exit		Strategy	Numeric computation *or expression*
	Offer	Interest	Technical problem		*Tactic*	Geometric expression *or link to drawing*
	Request	Risk-taking	Scripted facilitation		Perform	Algebraic expression but not a move below:
	Regulate	Resource	Unscripted facilitation		*Result*	Import new math info
	Repair typing	Norms	Drawing facilitation		Check	*Import and apply new math info*
	Respond, *more general than the codes below that are tied to problem solving*:	Home	*Contact facilitator*		*Corroborate/ counter*	Similar problem
	Follow	School			*Clarify [Request]*	Simplified case
	Elaborate	Collaborate group			Reflect	Infer pattern
	Extend	Collaborate individual			*Restate*	Divide in sub problems
	Setup	Sustain climate			Summarize	Test for boundaries
	Agree	Greet				Estimate
	Disagree					Trail and error
	Critique					Conduct unit analysis
	Explain					Work backwards
						Combinatoric

1. Segmentation: the entire log is coded and the unit of analysis is chat line as defined by the user in the chat.
2. Dimensions: each line is always given a conversational code and it can have 0 or 1 code from each of the other coding dimensions
3. Analysis: codes in single dimensions and combinations of codes in different dimensions can be considered together.
4. In general the 10 preceding lines can be used to guide threading and coding decisions. This does not apply to decisions whether the content of an utterance is new, nor for determining whether an utterance it is a setup, extension or elaboration, nor for determining if it restates content.
5. For background documentation and our coding scheme refer to http://mathforum.org/wiki/VMT?ThreadAnalResults

Acknowledgements

The authors express thanks to Murat Cakir, Nan Zhou, Ramon Toledo and Johann Sarmiento for their hard work in developing, testing and using the coding system in the reliability trails.

The reported research was funded by NSF grant no. 0325447 (Catalyzing and nurturing online workgroups to power virtual learning communities) and NSDL grant no. 0333493 (Collaboration services for the Math Forum digital library).

Table 1. Threading reconstruction (derived from reliability trail R1).

Line	Name	Utterance	Time	Delay	T1	T2	T3	TA
154	AME	How about you fir	7:28:03	0:15				
155	AME	Do you agree	7:28:35	0:32	154		154	154
156	AME	nvm	7:28:50	0:15				
157	MCP	I used cos(22.5) instead of .924. Got 4.2498ish	7:28:55	0:05	151	153	153	153
158	AME	lets go on	7:28:55	0:00	156	156	156	156
159	AME	Its close enough	7:29:16	0:21	157	157	157	157
160	AME	How about 4.25?	7:29:22	0:06			157	157
161	MCP	I guess use 4.6^ - 4.25^ to get BV^2	7:29:53	0:31	160	160		160
162	AME	ya	7:30:03	0:10	161	161	161	161
163	MCP	Then 16 * that, again	7:30:05	0:02		161	161	161
164	AME	I got 1.76 or so	7:31:03	0:58			161	
165	MCP	yes	7:31:09	0:06	164	164	164	164
166	AME	So the perimeter should be 28.16	7:31:28	0:19		164	164	164
167	FIR	ye!	7:31:44	0:16	166	164	166	166
168	FIR	*YES!	7:31:51	0:07	167	167	167	167

T1 = Thread coder 1, T2 = Thread coder 2, T3 = Thread coder 3, TA = Agreed after discussion.

Table 2. Conversational threading proportion agreement by coder pair and trail.

	R1			R2		
Pair	% thread	% same	% different	% thread	% same	% different
1 – 2	.832	.678	.268	.833	.653	.287
1 – 3	.778	.634	.272	.822	.662	.251
2 – 3	.750	.650	.313	.831	.668	.269

Table 3. Problem solving threading proportion agreement by coder pair and trial.

	R1			R2		
Pair	% thread	% same	% different	% thread	% same	% different
1 – 2	.756	.686	.143	.940	.934	.016
1 – 3	.805	.749	.120	.910	.894	.032
2 – 3	.753	.702	.109	.879	.854	.060

Table 4. Proportion and kappa by coder, trial and unit pool for conversation and social.

	Conversation dimension			
	R1 (*U = 500*)		R2 (*U = 450*)	
Pair	% agreement	Kappa	% agreement	Kappa
1 – 2	.750	.723	.735	.703
1 – 3	.644	.583	.724	.687
2 – 3	.692	.663	.724	.689

	Social dimension									
	R1					R2				
	12 codes			13 codes (*U = 500*)		12 codes			13 codes (*U = 450*)	
Pair	%	K	*U*	%	K	%	K	U	%	K
1 – 2	.550	.464	*209*	.812	.651	.646	.565	*180*	.857	.755
1 – 3	.495	.382	*218*	.788	.594	.543	.444	*162*	.835	.669
2 – 3	.529	.413	*187*	.824	.637	.506	.407	*164*	.820	.634

Table 5. Proportion and kappa by coder, trial and unit pool for support and problem solving.

	Support dimension									
	R1					R2				
	7 codes			8 codes (*U = 500*)		7 codes			8 codes (*U = 450*)	
Pair	%	K	*U*	%	K	%	K	U	%	K
1 – 2	.785	.747	*28*	.988	.876	.814	.701	*42*	.977	.858
1 – 3	.560	.526	*30*	.974	.716	.725	.647	*40*	.975	.846
2 – 3	.700	.661	*24*	.986	.825	.814	.747	*43*	.982	.891

	Problem solving dimension									
	9 codes			10 codes (*U = 500*)		11 codes			12 codes (*U = 450*)	
Pair	%	K	*U*	%	K	%	K	U	%	K
1 – 2	.469	.382	*177*	.821	.622	.657	.588	*178*	.864	.766
1 – 3	.351	.229	*168*	.782	.514	.553	.484	*197*	.804	.675
2 – 3	.439	.339	*148*	.834	.618	.556	.485	*187*	.815	.688

Figure 1

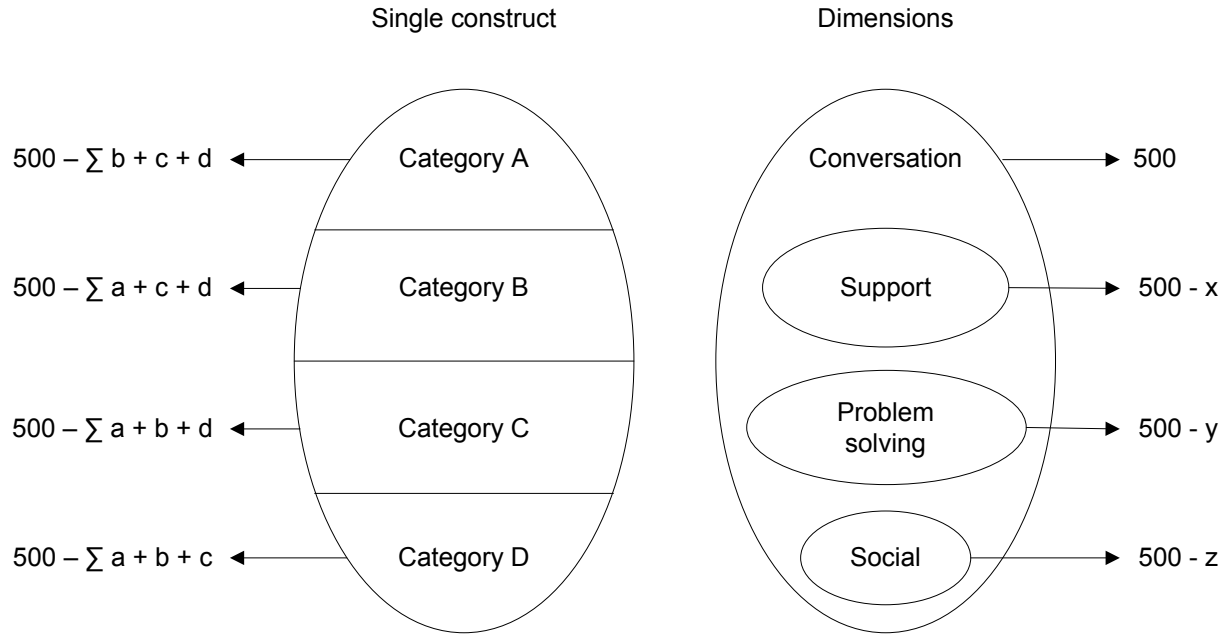
Single construct
Dimensions
500 – ∑ b + c + d
Category A
500 – ∑ a + c + d
Category B
500 – ∑ a + b + d
Category C
500 – ∑ a + b + c
Category D
Conversation
500
Support
500 - x
Problem solving
500 - y
Social
500 - z

Figure Caption

Figure 1. Pool of valid units for main categories in a single construct scheme versus dimensions.