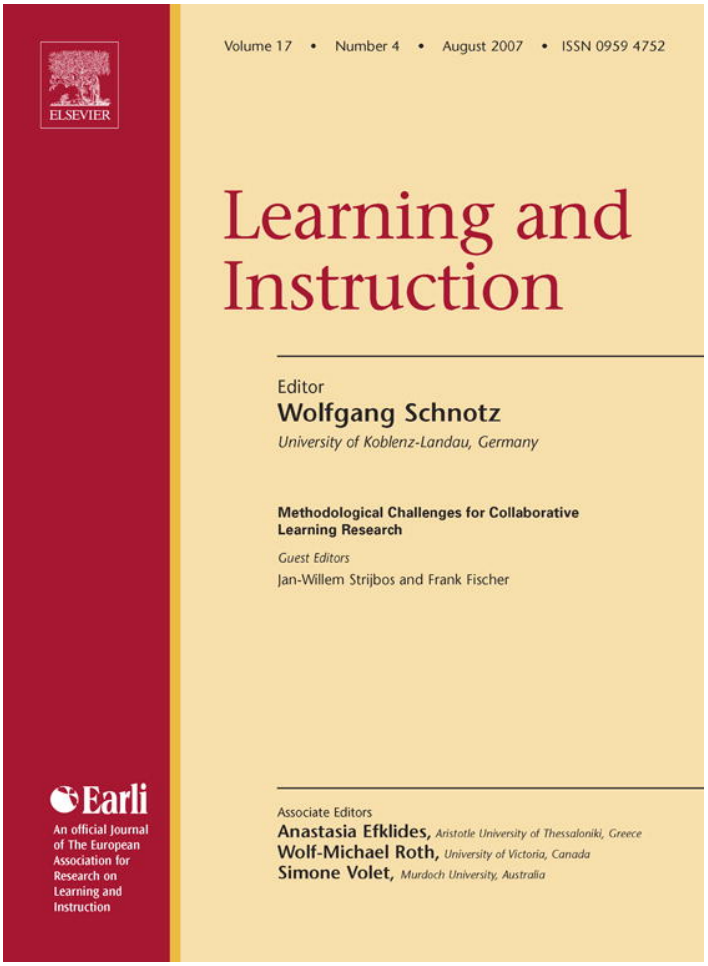

ELSEVIER

Learning and Instruction 17 (2007) 394–404

Learning and Instruction

www.elsevier.com/locate/learninstruc

Methodological issues in developing a multi-dimensional coding procedure for small-group chat communication

Jan-Willem Strijbos [a,*], Gerry Stahl [b]

[a] *Leiden University, Faculty of Social and Behavioural Sciences, Centre for the Study of Learning and Instruction, Section for Educational Studies, Department of Pedagogical Sciences, Wassenaarseweg 52, P.O. Box 9555, 2300 RB, Leiden, The Netherlands*

[b] *Drexel University, College of Information Science and Technology, Philadelphia, PA, USA*

Abstract

In CSCL research, collaboration through chat has primarily been studied in dyadic settings. This article discusses three issues that emerged during the development of a multi-dimensional coding procedure for small-group chat communication: (a) the unit of analysis and unit fragmentation, (b) the reconstruction of the response structure and (c) determining reliability without overestimation. Threading, i.e. connections between analysis units, proved essential to handle unit fragmentation, to reconstruct the response structure and for reliability of coding. In addition, a risk for reliability overestimation was identified. Implications for analysis methodology in CSCL are discussed.

Keywords: Content analysis; Methodology; Reliability; Threading; Coding

Coding of communication processes (content analysis) to determine effects of computer-supported collaborative learning (CSCL) has become a common research practice (Barron, 2003; Fischer & Mandl, 2005; Webb & Mastergeorge, 2003). In the past decade, research on CSCL has opened new theoretical, technical and pedagogical avenues of research. Comparatively less attention has, however, been directed to methodological issues associated with coding (Strijbos, Kirschner, & Martens, 2004).

Early attempts to analyse communication in computer-supported environments focused on counting messages to determine students' participation and on mean number of words as an indicator for the quality of messages. Later, methods like 'thread-length' analysis and 'social network analysis' expanded this surface-level repertoire. Now the CSCL research community agrees that surface methods can provide a useful initial orientation, but believes that more detailed analysis is needed to understand the underlying mechanisms of group interaction.

Content analysis is widely applied in collaborative learning research (see Barron, 2003; Gunawardena, Lowe, & Anderson, 1997; Schellens & Valcke, 2005; Strijbos, Martens, Prins, & Jochems, 2006; Weinberger & Fischer, 2006). Communication is segmented into analysis units (utterances), coded and their frequencies used for comparisons and/or statistical testing. Increasingly, collaborative learning studies are moving to a mixed-method strategy

* Corresponding author. Tel.: +31 71 5274048; fax: +31 71 5273619.
E-mail address: jwstrijbos@fsw.leidenuniv.nl (J.W. Strijbos).

0959-4752/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.learninstruc.2007.03.005

(Barron, 2003; Hmelo-Silver, 2003; Strijbos, 2004) and new techniques are being combined with known ones, such as multilevel modelling of content analysis data (Chiu & Khoo, 2003).

At present, however, the number of studies reporting on the specifics of an analysis method in detail is limited. With respect to content analysis this is highlighted by how many citations still reference Chi (1997), whose article was until recently the most cited article regarding the methodological issues involved. Within the CSCL community an academic discourse is gradually developing on issues such as analysis scheme construction, comparability and re-use (De Wever, Schellens, Valcke, & Van Keer, 2006), unit of analysis (Strijbos et al., 2006) and specific processes like argumentative knowledge construction (Weinberger & Fischer, 2006)—but many issues remain.

1. Background

This article reports on an attempt to use coding under circumstances that may be typical in CSCL research, but where coding has not generally been applied. The theory behind our research focuses on group processes and the meaning making that takes place in them. It is elaborated in Stahl (2006a) and Stahl, Koschmann, and Suthers (2006). The theory there recommends ethnomethodologically informed conversation analysis as the most appropriate analysis methodology, but we wanted to try to apply a coding approach as well.

Coding is most frequently used to compare research groups under controlled experimental conditions with well-defined dependent variables; we wanted to use coding to help us explore initial data where we did not yet have explicit hypotheses. Coding is often used in cases of face-to-face talk (e.g., in a classroom) or between communicating dyads; we were interested in online text-based synchronous interaction within small groups of three to five students. Educational and psychological research using coding generally takes utterances or actions of individuals as the unit of analysis; we wanted to focus on the small group as the unit of agency and identify group processes. In undertaking our inquiry into the use of coding under these circumstances, we strove for both reliability and validity. In this article, we take a close look at reliability and address issues of validity in our discussion.

Our test site, the VMT project (http://mathforum.org/vmt/), is developing an online service for students to engage in math discourse at a distance. This project takes a design-based research approach (Stahl, 2006b). It started very simply with a well-known technology (AOL IM®) and the established Math Forum Problem of the Week (PoW) service. The PoW service targets students in grades 3 through 12. It provides 'creative, non-routine challenges' for volunteers around the globe. The service is divided into four separate branches: algebra, geometry, pre-algebra and math fundamentals. The reported work with the coding scheme was conducted at the end of the first year of the 5-year research project (for examples of the problems see http://mathforum.org/pow/vmt/allproblems0304.html).

We wanted to understand what was happening in the chats along a number of dimensions. We wanted insights that would help us to develop the environment and the pedagogical approach. In particular, we were interested in how the students communicated, interacted and collaborated. We were also interested in how they engaged in math problem solving as a group. So we drew upon coding schemes from the research literature that addressed these dimensions while developing the VMT coding scheme.

2. VMT coding scheme

Multi-dimensional coding schemes are not a novelty in CSCL research, but they are often not explicitly defined. Henri (1992) distinguishes five dimensions: participation, social, interactive, cognitive, and metacognitive. Fischer, Bruhn, Gräsel, and Mandl (2002) define two dimensions: the 'content' and 'function' of utterances (speech acts). Finally, Weinberger and Fischer (2006) use four dimensions: participation, epistemic, argument, and social. These studies assign a single code to an utterance, or they code multiple dimensions that differ in the unitisation grain size (i.e., message, theme, utterance, sentence, etc.).

The first step in the development of the coding scheme was to determine the unit of analysis; its granularity can affect accuracy of coding (Strijbos et al., 2006). We decided to use the chat line as the unit of analysis mainly because it is defined by the user. It allowed us to avoid segmentation issues based on our (researcher) view. We empirically saw that the chat users tended to only 'do' one thing in a given chat line. Exceptions requiring a separate segmentation procedure were rare and too insubstantial to affect coding. We decided to code the entire log, including automatic system entries. In contrast to other multi-dimensional coding schemes unitisation is the same for all

dimensions: a chat line receives either a code or no code in each dimension—this allows for combinations of dimensions and expands the analytical scope.

We decided to separate communicative and problem-solving processes and conceptualised these as independent dimensions. Our initial scheme consisted of the conversational thread (who replies to whom), the conversation dimension (based on Beers, Boshuizen, Kirschner, & Gijselaers, 2005; Fischer et al., 2002; Hmelo-Silver, 2003), the social dimension (based on Renninger & Farra, 2003; Strijbos, Martens, Jochems, & Broers, 2004), the problem-solving dimension (based on Jonassen & Kwon, 2001; Polya, 1985), the math-move dimension (based on Sfard & McClain, 2003) and the support dimension (system entries and moderator utterances).

Then we spent the summer trying to apply these codes to 10 chats that we had logged in Spring 2004. Naturally, we wanted our coding to be reliable, so we checked on our inter-rater reliability as we went along. Problems in capturing what was taking place of interest in the chats and in reaching reliability led us to gradually evolve our categories. As the dimensions became more complicated with sub-categories, it became clear that some of them should be split into new dimensions. We ended with the categories in Appendix A, and the additions during calibration trials have been italicised (the math move and support dimension are not discussed in the remainder of this article and therefore not shown).

It turned out that it was important to conduct the coding of the different dimensions in a certain order, and to agree on the coding of one dimension before moving on to consider others. In particular, determining the threading of chat in small groups is fundamental to understanding the interaction. For the participants, confusion about the threading of responses by other participants can be a significant task and source of problems (Fuks, Pimentel, & De Lucena, 2006; O'Neill & Martin, 2003). For researchers, the determination of conversational threading is the first step necessary for analysis (Cakir, Xhafa, Zhou, & Stahl, 2005). Agreement on the threading by the coders establishes a basic interpretation of the interaction. Then, individual utterances can be assigned to codes in a reliable way. In addition, we were interested in the math problem solving. So we also determined the threading of math argumentation, which sometimes diverged from the conversational threading, often by referring further back to previous statements of math resources that were now being made relevant. Determining the problem-solving threading required an understanding of the math being done by the students, and often involved bringing math expertise into the coding process.

In this article, we focus on three issues that emerged in our attempt to apply a coding scheme in preliminary stages of CSCL research:

(a) We tried to use the natural unit of the chat posting as our unit for coding. This rarely led to problems with multiple contents being incorporated in a single posting, but rather with a single expressive act being spread over multiple postings.
(b) The reconstruction of the chat's response structure was an important step in analysing a chat. We developed a conversation thread and a problem-solving thread to represent the response structure.
(c) The goal of acceptable reliability drove the evolution of the coding scheme. The calculation of reliability itself had to be adjusted to avoid overestimation for sparsely coded dimensions.

3. Unit fragmentation and response structure reconstruction

We started with the calibration of the conversation dimension and combined this with threading in a single analysis step, but quickly discovered that 'threading' actually consisted of two issues namely 'unit fragmentation' and 'reconstruction of the response structure'. Unit fragmentation refers to fragmented utterances by a single author spanning multiple chat lines. These fragments make sense only if considered together as a single utterance. Usually, one of these fragments is assigned a conversational code revealing the conversational action of the whole statement, and the remaining fragments are tied to the special fragment by using 'setup' and 'extension' codes. This reduces double coding. Table 1 provides an example of both codes: line 155 is an extension to 154 and together they are a 'request' and line 156 is a setup to line 158 forming a 'regulation'.

CSCL research on chat technology previously focused on dyadic interaction (e.g., research on argumentation; Andriessen, Baker, & Suthers, 2003), which poses few difficulties to determine who responds to whom. In contrast, the VMT's small-group chat transcripts revealed that the chain of utterances was problematic. A discussion forum uses a threaded format that automatically inserts a response to a message as a subordinate object in a tree structure, and in

Table 1
Threading reconstruction (derived from reliability trial 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.

a similar vein, a prefix is added to the subject header of an e-mail reply. Current chat technology has no such indicators identifying the chain of utterances. Moreover, while there is no confusion about the intended recipient in a dyadic setting (the other actor), students in small groups often communicate simultaneously, making it easy to loose track of to whom they should respond. Coding small-group conversation in a chat required the reconstruction of the response structure as shown in Table 1.

Delay between utterances proved to be important. For example, lines 157 and 158 fully overlap (no delay) and the delay between lines 166 and 167 of 16 s reveals that the short utterance of 167 is more likely to be connected 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.

Connecting utterances to handle unit fragmentation and to reconstruct the response structure is performed simultaneously, and referred to as 'threading'. The threading is performed separately from the conversational coding, including assignment of extension and setup, because not all spanned utterance connections concern fragmentation. There is one infrequent exception of a spanned utterance in the shape of three fragments coded as 'explain/critique' + 'elaborate' + 'extension', but this emphasises that coding of extend and setup should be performed separately. In other words, threading only reconstructs connections between the user-defined chat lines that form (a) a fragment of a spanned utterance or (b) a response to a previous utterance, but the nature of the chat line is decided during coding and not during threading. It also highlights that a coder should be familiar with the codes to ensure that s/he knows which lines should be considered for threading because the conversational code depends on whether or not a thread is assigned.

Calibration trials for the problem-solving dimension revealed a similar need for the reconstruction 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)—prior to the coding of problem solving.

Calibration trials showed that threading is of utmost importance for the analysis of chat-based small-group problem solving and should be assigned prior to the (conversational) coding. In the next section we will discuss the reliability for threading and coding of three dimensions in detail, as their calculation presented additional methodological issues—more specifically the risk for reliability overestimation. In line with Strijbos et al. (2006) we address reliability stability by presenting two trials, each covering about 10% of the data.

4. Reliability of threading, coding and reliability overestimation

4.1. Reliability of threading

Threading is already a deep interpretation of the data and therefore a reliability statistic should be determined. The calculation of 'threading reconstruction' reliability proved complicated, because coders can assign a thread indicator

to a chat line or not, assign an indicator to the same chat line or to a different chat line. As a result, only a proportion agreement can be computed. We used three coders (first author and two research assistants) and computed two indices for all possible dyads:

- For the assignment of a thread or not by both coders (% thread);
- For the assignment of the same thread whenever both assigned a thread (% same).

Table 2 presents 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 449 chat lines. The top of Table 2 presents the results for the conversational thread and the bottom the results for the problem-solving thread.

A threshold for the proportion agreement reliability of segmentation does not exist in CSCL research (De Wever et al., 2006; Rourke, Anderson, Garrison, & Archer, 2001), nor in the field of content analysis (Neuendorf, 2002; Riffe, Lacy, & Fico, 1998). Given the various perspectives in the literature, a range of .70–.80 for proportion agreement can serve as the criterion value. Combined results for the conversational thread reveal that, on average, both coders assign a thread in 80.7% of all cases. Overall, 72.2% of the thread assignments are the same. These combined results show that the reliability of conversational threading is actually quite stable and fits the .70–.80 range.

The results of both reliability trials reveal for the problem-solving thread that, on average, in 87% of all the instances both coders assigned a thread. Of all threading assignments by either coder 91.5% are the same. These results show that the reliability of problem-solving threading exceeds the .70–.80 range. It should be noted that the problem-solving thread is often the same as the conversation thread, so the reliability indices are automatically higher. The R2 selection also contained fewer problem-solving utterances than R1, so the problem-solving thread is more similar to the conversational thread and thus reliability higher. Since the reliability of problem-solving threading depends on the number of utterances that actually contain problem-solving content, it will fluctuate between transcripts. Therefore, the first trial should be regarded as a satisfactory lower bound: 77.1% for thread assignment and 89.9% for same thread assignment.

4.2. Reliability of three coding dimensions and reliability overestimation

Given the impact of the conversational and problem-solving threads during the calibration sessions, codes were added or changed, definitions adjusted, prototypical examples added, and rules to handle exceptions established. Nine calibration trials were conducted prior to the reliability trials. We used three coders (first author and two research assistants) and adopted a stratified coding approach for each reliability trial: the coders first individually assigned the conversation threads, followed by a discussion to construct an agreed upon conversational thread, after which each coder independently coded the conversational and social dimension. Next, coders first individually assigned the problem-solving thread before a discussion was held to construct an agreed upon problem-solving thread, followed by assigning the problem-solving codes. Between both reliability trials, minor changes were made in the wording of a definition or adjusting a rule. Mastery of the coding procedure is laborious. Per dimension, it takes about 20 h of training and discussion with an experienced coder.

Table 2
The proportion agreement indices for the conversational and problem-solving thread by coder pair and reliability trial

Pair	R1		R2	
	% Thread	% Same	% Thread	% Same
Conversational thread				
1–2	.832	.731	.835	.712
1–3	.778	.727	.824	.749
2–3	.750	.687	.832	.730
Problem-solving thread				
1–2	.756	.928	.942	.983
1–3	.805	.879	.909	.967
2–3	.753	.890	.880	.935

In contrast to our initial conceptualisation of the dimensions as being independent we have been thus far unable to avoid ties between some of the conversational codes and the problem-solving dimension. Coding qualitatively different processes, social versus problem-solving, using the same data corpus was problematic—foremost regarding 'elaborate', 'explain' and 'critique' categories. The implications of ties for the validity of the coding scheme should be determined, but this is beyond the scope of the current article.

Calculating the reliability for the conversation, social, and problem-solving dimensions proved to be less straightforward than expected. Each chat line receives a conversation code and can have either one or no code for any other dimension, but not all chat lines are eligible to receive a particular code. The social and problem-solving dimensions only apply to a portion of all of the chat lines, and the pool of valid units will fluctuate between different pairs of coders. When not all units are eligible to receive a code we should decide how we handle units coded by only one coder and the units not coded by both coders in the reliability computation:

(a) Include only units coded by both coders (exclude units with missing values);
(b) Categorise missing values as 'no code' and include this category;
(c) Categorise missing values and non-coded units as 'no code' and include this category.

For possibilities (a) and (c) we calculated three reliabilities indices as suggested by De Wever et al. (2006): proportion agreement (%), Cohen's kappa (κ) and Krippendorffs alpha (α) (the latter two correct for chance agreement) for each dimension and pair of coders. Option (b) was only computed for kappa and alpha. To determine whether the reliability is sufficient the .70–.80 range is mostly used as criterion for proportion agreement. Perspectives in the literature on a criterion value for kappa differ, but in our opinion these criteria—intermediate, strict and lenient—apply best: below .45 'poor', .45–.59 'fair', .60–.74 'good', and .75 and above 'excellent' (De Wever et al., 2006; Landis & Koch, 1977; Neuendorf, 2002). We apply the same criteria to alpha. Table 3 shows the reliability results for the conversation, social and problem-solving dimension.

Although proportion agreement is still often used, it is insufficient to serve as an indicator for reliability because it does not correct for chance agreement, and we report this solely for comparison. Kappa is computed because this is the most widely used statistic that corrects for agreement by chance. However, recent publications revealed that kappa behaves strange, i.e. the kappa for two coders with a radically different distribution of frequencies over categories will be higher than coders with a similar distribution (Artstein & Poesio, 2005; Krippendorff, 2004). Alpha does not suffer from this statistical artefact, so it should be preferred. We retain kappa for comparison because alpha is not widely used in CSCL or educational research. We will first discuss the pair-wise comparisons for the social and problem-solving dimension.

When only those units coded by both coders are included in the computation—κ_1 and α_1—the reliability is consistently higher than proportion agreement, which is expected because κ_1 and α_1 do not treat all units coded by only one coder as disagreement. It should be noted that alpha affords to 'include' missing values in the data matrix, however, units coded by only one coder are ignored in the final computation. So, although it seems that more units are included there is computationally no difference with the case where these units are excluded (Table 3 shows the number of units that 'appear' to be used for the computation for α_1 but they are in reality the same as for κ_1).

When the missing values for units that were coded by only one coder are categorised 'no code' and this 'extra' category is included in the computation—κ_2 and α_2—reliability drops. This is stronger for the social dimension as compared to the problem-solving dimension, and is caused by the number of missing values; more missing values lead to a stronger downward correction when these are treated as disagreement. Alpha and kappa have similar values, but differ slightly (caused by the different distribution of frequencies over categories).

When the missing values and all units that were not coded by both coders are included and categorised as 'no code'—$\%_A$, κ_A and α_A—proportion agreement is consistently higher, α_A is higher than α_2 for the social and problem-solving dimension but is lower than α_1 for the social dimension and equal to α_1 for the problem-solving dimension. The same pattern is visible for the three kappa indices.

Since proportion agreement does not correct for chance agreement and kappa suffers from a statistical artefact, alpha is preferred. Excluding missing values in the computation neglects a source of disagreement and inflates reliability, so α_1 is not adequate. Including all units that were not coded by both coders appears appealing and consistent but treats those units that are conceptually not eligible to receive a code as agreement. So, α_A also inflates

Table 3
Proportion agreement, kappa and alpha by coder for the conversation, social and problem-solving dimension

Conversation dimension

Pair	R1 ($U = 500$)			R2 ($U = 449$)		
	%	κ	α	%	κ	α
1−2	.750	.723	.704	.735	.703	.702
1−3	.644	.583	.600	.724	.687	.686
2−3	.692	.663	.654	.724	.689	.681
Three coders			.653			.689

Social dimension

Pair	R1								R2							
	Missing excluded			Missing as 'no code'		Missing and no-code units included ($U = 500$)			Missing excluded			Missing as 'no code'		Missing and no-code units included ($U = 449$)		
	%	κ_1	α_1	κ_2	α_2	$\%_A$	κ_A	α_A	%	κ_1	α_1	κ_2	α_2	$\%_A$	κ_A	α_A
1−2	.550	.835	.850	.464	.430	.812	.651	.641	.646	.748	.733	.565	.550	.857	.755	.733
	208	*127*	*208*	*208*	*208*				*176*	*140*	*176*	*176*	*176*			
1−3	.495	.793	.771	.382	.372	.788	.594	.593	.543	.737	.733	.444	.412	.835	.669	.649
	218	*129*	*218*	*218*	*218*				*163*	*107*	*163*	*163*	*163*			
2−3	.529	.798	.831	.413	.439	.824	.637	.656	.506	.730	.739	.407	.367	.820	.634	.609
	185	*115*	*185*	*185*	*185*				*174*	*106*	*174*	*174*	*174*			
Three coders			.787		.462			.629			.735		.480			.668
			225		*225*						*182*		*182*			

Problem-solving dimension

Pair	R1								R2							
	Missing excluded			Missing as 'no code'		Missing and no-code units included ($U = 500$)			Missing excluded			Missing as 'no code'		Missing and no-code units included ($U = 449$)		
	%	κ_1	α_1	κ_2	α_2	$\%_A$	κ_A	α_A	%	κ_1	α_1	κ_2	α_2	$\%_A$	κ_A	α_A
1−2	.469	.631	.628	.382	.385	.821	.622	.613	.657	.674	.666	.588	.576	.864	.766	.762
	178	*127*	*178*	*178*	*178*				*178*	*158*	*178*	*178*	*178*			
1−3	.351	.564	.543	.229	.242	.782	.514	.504	.553	.649	.662	.484	.464	.804	.675	.665
	172	*97*	*172*	*172*	*172*				*195*	*147*	*195*	*195*	*195*			
2−3	.439	.542	.520	.339	.340	.834	.618	.608	.556	.576	.654	.485	.469	.815	.688	.667
	148	*106*	*148*	*148*	*148*				*190*	*146*	*190*	*190*	*190*			
Three coders			.563		.370			.576			.650		.523			.699
			181		*181*						*196*		*196*			

% = percentage agreement; κ = Cohen's kappa; α = Krippendorff's alpha; κ_1 = kappa with missing excluded; α_1 = alpha with missing excluded; κ_2 = kappa with missing as disagreement; α_2 = alpha with missing as disagreement; $\%_A$, κ_A, and α_A = percentage, kappa and alpha when all units are included. The number of eligible (valid) units is shown in italics.

reliability and is not adequate. Including only those units coded by either coder, categorising missing values as 'no code', is the most strict computation. Thus, α_2 should be preferred although this statistic is a slight underestimation of the possible 'eligible' units—because it ignores the ambiguous units that both coders considered but did not code—but this is favoured given the substantial overestimation if missing values are excluded or all non-coded units are included.

The pair-wise comparisons provide insight into the performance of particular coders, but if more than two coders are available this should be preferred. We had three coders and alpha is suited to compute reliability for more than two coders (although Fleiss kappa can also correct for multiple coders it applies only to nominal data, alpha can also be used for ordinal, interval and ratio data). Again, α_2 is preferred over α_1 and α_A for the case of three coders, and appears the best approximation for the reliability for the social and problem-solving dimension.

Considering the reliability statistics for three coders, alpha for the conversation dimension can be considered 'good' for both trails, .653 for R1 and .689 for R2. The alpha for the social dimension can be considered 'fair' for both trials, .462 for R1 and .480 for R2. The alpha for the problem-solving dimension is 'poor' for R1 (.370) and 'fair' for R2 (.523).

5. Discussion

CSCL research using chat technology has focused primarily on dyads. The VMT project investigates chat-based small-group problem solving. During the development of a multi-dimensional coding scheme to analyse interactions in these groups, three new issues emerged that have strong implications for content analysis methodology and practice in general and chat communication in particular.

The first methodological issue concerns unit fragmentation. We chose the chat line as the unit of analysis because this is defined by the user, but frequently an utterance spanned across several chat lines makes sense only when considered as a whole. Consequently, connections between these units were required prior to coding, and two codes were added to the conversation dimension to mark these fragments (setup and extension).

The second issue concerns the need to reconstruct the response structure. Whereas in a dyadic chat the intended recipient is always the other partner, it is not easy to determine this in a small group. Similarly to fragmentation, the connection between chat lines forming a chain of responses needs to be reconstructed prior to coding of the conversation dimension. Furthermore, the delay between chat line postings proved to be relevant to determining the response structure. Also, a coder must be familiar with the conversational codes. Assignment of both types of connections is performed simultaneously and termed 'threading' and a deep interpretation of what is going on in the chat. Aggregating all coding divergence would result in very low reliabilities, so agreement on threading prior to coding is necessary.

The third methodological issue concerns reliability calculation. We conducted two trials and computed the reliability for both types of threading. Reliability for the conversation and problem-solving threading could only be expressed as a proportion agreement, but this proved to be sufficiently reliable. Calculation of reliability for the social and problem-solving dimension was problematic: not all chat lines are valid analysis units for these dimensions and can lead to overestimation of their reliability. The extent of overestimation was shown by calculating reliability for the case where (a) only units coded by both coders are included (missing values are excluded), (b) missing values are categorised as 'no code' and included in the computation, and (c) missing values and non-coded units are categorised as 'no code' and included in the computation. We computed and compared three reliability indices and concluded that excluding missing values and including all non-coded units lead to overestimation. Including missing values as a 'no code' category is the most strict computation and a slight underestimation of the reliability. In our opinion a slight underestimation should be favoured given a substantial overestimation if units with missing values are excluded or all non-coded units are included. If available the use of more than two coders is preferred, and the valid pool of units should be reported (see for example Hurme & Järvelä, 2005, p. 6).

We included proportion agreement and Cohen's kappa for comparison, although both statistics are problematic. Overall, coding reliability—Krippendorff's alpha for three coders—ranged 'poor' to 'good' in the first trial and 'fair' to 'good' in the second trail. Nevertheless, reliability is only one aspect of a coding scheme—addressing the extent to which the coding can be reproduced—and it should not be mistaken for validity. We conclude with some reflections on validity. Once we had reliable coding of 10 chat logs, we looked for statistical patterns. It turned out that the chats almost fell into two sets depending upon whether the students had seen the math problems in

advance of their chats or not. However, there were two anomalous chats that fell into the wrong sets. The use of codes brought this anomaly to our attention, but could not explain it. Using conversation analysis, we could see a difference in interaction patterns that we termed expository versus exploratory (Mercer & Wegerif, 1999; Zemel, Xhafa, & Stahl, 2005). Subsequently, we found that students working in our chat environment developed methods of interacting that were not adequately captured—let alone explained—by codes adopted from the work of researchers investigating other media or from a priori theories of interaction. For instance, we determined that 'math proposal adjacency pairs' often play a distinctive driving role in our math chats (Stahl, 2006c). Ethnomethodologically informed design-based research needs to grasp the methods that participants creatively invent in response to innovative learning situations and technologies; they cannot simply reduce everything to instances of categories of actions generalised from past studies.

Also, we are particularly interested in group cognition (Stahl, 2006a) that takes place at the group unit of analysis, while coding schemes generally focus on the individual. For instance, we look at problem solving by the group as a whole (Stahl, 2006d). Our coding scheme tried to capture group phenomena like proposal bid-and-uptake or interaction question-and-answer by coding these as sequences of individual contributions (e.g., offer followed by response). The format of chat logs and the traditions of coding practice misled us to fragment group interactions into individual contributions. We now want to look at paired interactions and longer sequences as atomic elements of chats.

As the VMT environment evolved and incorporated a shared whiteboard, graphical referencing, math symbols and other functionality, even our multi-dimensional coding of utterances could not capture the increasingly complex and innovative interactions (Stahl, 2006e). To understand the unique behaviors as students adapt to the new environment—custom technology, pedagogical guidance, open-ended math worlds—we need to look closely at the design of unique group interactions, and not simply code them with pre-existing categories, no matter how multi-dimensional and reliable. While general codes can be applied to many of these phenomena, they do not capture what is new, as required for design-based research. Reducing the chat to a sequence of codes that are general enough to be applied reliably, can eliminate the content and details that are of particular interest (Stahl, 2002). This is a paradox of reliable and valid coding efforts in exploratory CSCL research.

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 trials. The reported research was funded by NSF grants 0325447 and 0333493.

Appendix A. VMT coding scheme example (italic signals addition during calibration)

C-thread	Conversation	Social	*PS-thread*	Problem solving
Reply to U_i	No code	Identity self	*Connect to* U_i	Orientation
	State	Identity other		Strategy
	Offer	Interest		*Tactic*
	Request	Risk-taking		Perform
	Regulate	Resource		*Result*
	Repair typing	Norms		Check
	Respond, *more general than the codes below that are tied to problem solving*:	Home		*Corroborate/counter*
	Follow	School		*Clarify*
	Elaborate	Collaborate group		Reflect
	Extend	Collaborate individual		*Restate*
	Setup	Sustain climate		Summarise
	Agree	Greet		
	Disagree			
	Critique			
	Explain			

References

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

Artstein, R., & Poesio, M. (2005). *Kappa^3 = alpha (or beta)* (NLE Technical Note 05–1). University of Essex: Natural Language Engineering and Web Applications Group.

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. (2005). Computer support for knowledge construction in collaborative learning environments. *Computers in Human Behavior, 21*, 623–643.

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.

Chi, M. T. H. (1997). Quantifying qualitative analysis of verbal data: a practical guide. *The Journal of the Learning Sciences, 6*, 271–315.

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 Wever, B., Schellens, T., Valcke, M., & Van Keer, H. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: a review. *Computers & Education, 46*, 6–28.

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

Fischer, F., & Mandl, H. (2005). Knowledge convergence in computer-supported collaborative learning: the role of external representation tools. *The Journal of the Learning Sciences, 14*, 405–441.

Fuks, H., Pimentel, M., & De Lucena, C. J. P. (2006). R-U-Typing-2-Me? Evolving a chat tool to increase understanding in learning activities. *International Journal of Computer-Supported Collaborative Learning, 1*, 117–142.

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.

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

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.

Krippendorff, K. (2004). Reliability in content analysis: some common misconceptions and recommendations. *Human Communication Research, 30*, 411–433.

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

Mercer, N., & Wegerif, R. (1999). Is "exploratory talk" productive talk? In K. Littleton, & P. Light (Eds.), *Learning with computers: Analyzing productive interaction* (pp. 79–101). New York, NY: Routledge.

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

O'Neill, J., & Martin, D. (2003). *Text chat in action*. Paper presented at the ACM Conference on Groupware (GROUP 2003), Sanibel Island, FL, USA.

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

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.

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

Stahl, G. (2002). Rediscovering CSCL. In T. Koschmann, R. Hall, & N. Miyake (Eds.), *CSCL 2: Carrying forward the conversation* (pp. 169–181). Hillsdale, NJ: Lawrence Erlbaum Associates.

Stahl, G. (2006a). *Group cognition: Computer support for building collaborative knowledge*. Cambridge, MA: MIT Press.

Stahl, G. (2006b). *Virtual Math Teams project: An overview of VMT*. Retrieved January 23, 2007, from: http://www.mathforum.org/vmt/researchers/orientation.html.

Stahl, G. (2006c). Sustaining group cognition in a math chat environment. *Research and Practice in Technology Enhanced Learning (RPTEL), 1*, 85–113.

Stahl, G. (2006d). Analyzing and designing the group cognitive experience. *International Journal of Cooperative Information Systems (IJCIS), 15*, 157–178.

Stahl, G. (2006e). Supporting group cognitions in an online math community: a cognitive tool for small-group text referencing in chat. *Journal of Educational Computing Research, 35*, 103–122.

Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), *Cambridge handbook of the learning sciences* (pp. 409–426). Cambridge, UK: Cambridge University Press.

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

Strijbos, J. W., Kirschner, P. A., & Martens, R. L. (Eds.). (2004). *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. (2004). 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. (2006). Content analysis: what are they talking about? *Computers & Education, 46*, 29–48.

Webb, N. M., & Mastergeorge, A. M. (2003). The development of students' helping behaviour and learning in peer-directed small groups. *Cognition & Instruction, 21*, 361–428.

Weinberger, A., & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education, 46*, 71–95.

Zemel, A., Xhafa, F., & Stahl, G. (2005, September). *Analyzing the organization of collaborative math problem-solving in online chats using statistics and conversation analysis*. Paper presented at CRIWIG 2005 conference, Recipe, Brazil.