Chapter 22

A Multidimensional Coding Scheme for VMT

Jan-Willem Strijbos

JWStrijbos@FSW.leidenuniv.nl

Abstract: In CSCL research, collaboration through chat has primarily been studied in dyadic settings. In VMT's larger groups it becomes harder to specify procedures for coding postings because the interactions are more complicated and ambiguous. This chapter discusses four issues that emerged during the development of a multidimensional coding procedure for small-group chat communication: (a) the unit of analysis and unit fragmentation, (b) the reconstruction of the response structure, (c) determining reliability without overestimation, and (d) the validity of constructs inspired by diverse theoretical-methodological stances. 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 is illustrated. Implications for reliability, validity and analysis methodology in CSCL are discussed.

Keywords: Unit of analysis, response structure, reliability, validity, coding scheme, methodology

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 analyze 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 (Barron, 2003; Gunawardena, Lowe & Anderson, 1997; Schellens & Valcke, 2005; Strijbos et al., 2006; Weinberger, 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 (Barron, 2003; Hmelo-Silver, 2003; Strijbos, 2004) and new techniques are being combined with known ones, such as multilevel modeling of content analysis data (Chiu & Khoo, 2003; Cress, 2008).

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 et al., 2006), unit of analysis (Strijbos et al., 2006) and specific processes like argumentative knowledge construction (Weinberger, 2006)—but many issues remain.

Background

This chapter 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 reported work with the coding scheme was conducted at the end of the first year of the VMT Project.

The theory behind our research focuses on group processes and the meaning making that takes place in them, as elaborated by Stahl (2006a; Stahl, Koschmann & Suthers, 2006). The theory 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.

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 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. In this chapter, we take a close look at both reliability and validity of the coding scheme.

VMT Coding Scheme

The VMT coding scheme can be characterized as a multidimensional coding scheme. Multidimensional 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 meta-cognitive. Fischer, Bruhn, Gräsel & Mandl (2002) define two dimensions: the content and function of utterances (speech acts). Finally, Weinberger & 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 unitization 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-generated entries. In contrast to other multidimensional coding schemes unitization 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 conceptualized these as independent dimensions. Our initial scheme consisted of the conversational thread (who replies to whom), the conversation dimension based on (Beers et al., 2005; Fischer et al., 2002; Hmelo-Silver, 2003), the social dimension based on (Renninger & Farra, 2003; Strijbos, Martens et al., 2004), the problem-solving dimension based on (Jonassen & Kwon, 2001; Polya, 1945/1973), 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 ten 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 dimensions. As the dimensions became more complicated with sub-codes, it became clear that some of them should be split into new dimensions. We ended with the dimensions in Table 22-1, and the additions during calibration trials have been italicized (the math move and support dimension are not discussed in the remainder of this chapter 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 (see Chapter 21). For researchers, the determination of conversational threading is the first step necessary for analysis (see Chapter 20). 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 chapter, we focus on four 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 analyzing 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 over-estimation for sparsely coded dimensions.

(d) Irrespective of reliability we wanted to take advantage of the diverse theoretical-methodological stances within the VMT research team that best reflected behaviors of collective interest (*validity*).

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. Log 22-1 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".

Table 22-1. VMT coding steps (italic signals addition during calibration).

Step 1	Step 2	Step 3	*Step 4*	Step 5
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		Summarize
	Agree	Greet		
	Disagree			
	Critique			
	Explain			

CSCL research on chat technology previously mainly 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 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 Log 22-1.

Log 22-1.

Line	Time	Delay	Name	Utterance	T1	T2	T3	TA
154	7:28:03	0:15	AME	How about you fir				
155	7:28:35	0:32	AME	Do you agree	154		154	154

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

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

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 seconds 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 emphasizes 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 coders should be familiar with the codes to ensure that they know 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.

Reliability of Threading, Coding and Reliability Overestimation

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 (author and two research assistants) and computed two indices for all possible coder 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 22-2 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 22-2 presents the results for the conversational thread and the bottom the results for the problem-solving thread.

Table 22-2. The proportion-agreement indices.

	Conversational thread			
	R1		R2	
Pair	% thread	% same	% thread	% same
1 – 2	.832	.731	.835	.712
1 – 3	.778	.727	.824	.749
2 – 3	.750	.687	.832	.730

	Problem-solving thread			
	R1		R2	
Pair	% thread	% same	% thread	% same
1 – 2	.756	.928	.942	.983
1 – 3	.805	.879	.909	.967
2 – 3	.753	.890	.880	.935

A threshold for the proportion-agreement reliability of segmentation does not exist in CSCL research (De Wever et al., 2006; Rourke et al., 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 to .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 to .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 to .80 range. It should be

noted that the problem-solving thread is very 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 the reliability is 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.

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 (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. The final version of the coding scheme included 40 code definitions (with examples of actual data samples) in 5 dimensions (not counting the mathematical and system-support dimensions) (see Table 22-1). Mastery of the coding procedure is laborious; some dimensions take about twenty hours of training and discussion with an experienced coder.

In contrast to our initial conceptualization 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—especially involving "elaborate," "explain" and "critique" codes. The implications of ties for the validity of the coding scheme will be discussed in the section on validity.

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 or none in the reliability computation:

(a) Include only units coded by both coders (exclude units with missing values);
(b) Categorize missing values as "no code" and include this code;

(c) Categorize missing values and non-coded units as "no code" and include this code.

For possibilities (a) and (c) we calculated three reliability indices as suggested by De Wever et al. (2006): proportion agreement (%), Cohen's kappa (κ) and Krippendorff's alpha (α) for each dimension and each pair of coders.

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 strangely, i.e., the kappa for two coders with a radically different distribution of frequencies over categories will be higher than for coders with a similar distribution (Artstein & Poesio, 2005; Krippendorff, 2004). Alpha does not suffer from this statistical artifact, so it should be preferred. We retain kappa for comparison because alpha is not widely used in CSCL or educational research.

Option (b) was only computed for kappa and alpha. To determine whether the reliability is sufficient the .70 to .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 to .59 "fair", .60 to .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 22-3 shows the reliability results for the conversation, social and problem-solving dimension. 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 allows including 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 22-3 shows the number of units that appear to be used for the computation for α_1, although they are in reality the same as for κ_1. % = 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, analysis units in italics, $\%_A$, κ_A, and α_A = percentage, kappa and alpha when all units are included.)

Table 22-3. Proportion agreement, kappa and alpha.

	Conversation dimension															
	R1 (*U = 500*)								R2 (*U = 449*)							
Pair	%			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		
3 coders						.653								.689		

	Social dimension															
	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*)		
Pair	%	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*			
3 coders			.787		.462			.629			.735		.480			.668
			225		*225*						*182*		*182*			

	Problem-solving dimension															
	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*)		
Pair	%	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*			
3 coders			.563		.370			.576			.650		.523			.699
			181		*181*						*196*		*196*			

When the missing values for units that were coded by only one coder are categorized "no code" and this "extra" code 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 categorized 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 artifact, 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 reliability and is not adequate. Including only those units coded by either coder, categorizing missing values as "no code", is the strictest 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 favored 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).

Validity of the VMT Coding Scheme

Although the methodological debate in CSCL research has intensified over the past decade (Strijbos & Fischer, 2007), it is apparent that regarding content analysis the issue of reliability has received much more attention than validity and generalizability. Rourke & Anderson (2004) convincingly argued that content analysis should be regarded as a form of testing and measurement and stressed the

importance of validity, especially when the analysis moves from description to making inferences. Their approach to validity in content analysis is modeled on Messick's (1989; 1995) aspects of construct validity. Rourke & Anderson (2004) describe five steps for developing a theoretically valid protocol:

(a) Identifying the purpose of the coded data (content aspect),
(b) Identifying behaviors that represent the construct (substantial aspect),
(c) Reviewing the codes and indicators (structural aspect),
(d) Holding preliminary try-outs and
(e) Developing guidelines for administration, scoring and interpretation of the coding scheme.

We will first briefly discuss the development of the VMT coding scheme with respect to these five steps and elaborate on design decisions made, followed by some empirical evidence for validity. Finally, Messick's generalizability aspect and external aspect will be briefly discussed in view of the current state of content analysis literature in CSCL.

Identifying the Purpose of the Coded Data

As briefly stated in the background section, we were interested in understanding what was happening in the chats—how students communicated, interacted and collaborated—to obtain insights that would help us to develop the environment and the pedagogical approach. Thus, the purpose of the VMT coding scheme was to describe collaborative processes of small groups solving a mathematical problem via chat, rather than drawing inferences (or stated differently, hypothesis generation rather than hypothesis testing).

Identifying Behaviors that Represent the Construct

Our dimensions of interest—conversation, social and problem solving—are latent constructs and inferred from observable behaviors (utterances). Construct validity draws on the connection between theory and method. This requires careful operationalization of behaviors to avoid construct under-representation and construct-irrelevant variance (Messick, 1989; 1995). Or in other words, that the coding scheme "neither leaves out behaviors that should be included, nor includes behaviors that should be left out" (Rourke & Anderson, 2004, p. 9).

Given the exploratory focus and descriptive purpose of coding we adopted a broad perspective on processes of interest. While developing the VMT coding scheme we relied on diverse theoretical-methodological stances within the research team, i.e., quantitative content analysis and qualitative approaches such as conversation analysis and ethnographic perspectives (e.g., grounded theory). We wanted to take advantage of these different viewpoints to construct a coding scheme that best reflected behaviors that we were collectively interested in. The codes of the scheme are based on literature study (published coding schemes) and transcript observations. They reflect the different theoretical approaches: speech act (e.g.,

"offer", "agree" and "disagree"), conversation analysis (e.g., "repair typing") and grounded theory (e.g., "follow" and "sustain social climate").

With its combined theoretical-methodological perspective the coding scheme can be regarded as an example of hybrid analysis methodologies called for by Suthers (2005). As the development of hybrid methodologies induces theoretical boundary-crossing, the question arises whether internal validity (relevant behaviors by participants from a single theoretical perspective) takes precedence over the substantial aspect of validity (relevant behaviors by participants from a combination of theoretical perspectives). In other words, a combination of theoretical perspectives appears more susceptible to construct-irrelevant variance, whereas a single theoretical perspective appears more susceptible to construct under-representation. In our view, hybrid analysis methodologies are well suited for hypothesis generation and descriptive analyses. Although we acknowledge the risk of construct-irrelevant variance, they do not automatically result in bias invalidating the outcomes of exploratory analyses, but can reveal new possible ways to describe the data.

Reviewing the Codes and Indicators

A provisional coding scheme was constructed by a researcher experienced in content analysis (author). The coding scheme was then discussed with three senior VMT researchers with diverse theoretical-methodological backgrounds: conversation analysis, ethnography and mathematical problem solving. We conducted three discussion rounds where codes and indicators were added and deleted, while trying to balance the diverse perspectives on interaction analysis and the behaviors of interest. In between discussions we applied the codes to transcript excerpts (individually and in pairs) moving back and forth between the codes, definitions, indicators, the data and reasoning about it. The experiences were discussed in the following meeting and the codes adapted accordingly. The coding scheme evolved from each utterance receiving a single code to a coding scheme in which each utterance receives more than one code—but each of them in a separate dimension.

The tension between the theoretical-methodological stances was reflected strongest in the discussion on the number of codes and the degree of specificity needed to describe behaviors of interest. The debate focused on the desire for a parsimonious set of codes versus inclusion of all relevant—even if infrequent—behaviors. A point in case are the codes "school" and "home". They are relevant from an interactional point of view because VMT participants only met online and references to their school or home context can be indicative for the social climate in the group, but their infrequent occurrence makes these codes more suited for descriptive analyses rather than statistical inferences.

Interestingly, the issue of the number of codes has so far not been explicitly addressed in leading publications on content analysis and in CSCL research. Obviously a set rule for the number of codes does not make much sense, but there are several aspects that can guide this decision: level of detail required, theory-driven versus a data-driven focus (or in other words researcher codes versus participant codes), cognitive demand of coding (a large amount of codes is cognitively more

demanding and increases the risk of errors due to fatigue), and representativeness of the behavior of interest. Given these issues we initially decided to limit the number of codes in each dimension to a maximum of 12. Only the conversation dimension was further expanded to 15 codes during calibration.

Finally, there were utterances that could not be assigned to any of the codes. Often "no code" is used to handle the utterances that do not appear to fit any of the codes in the coding scheme. Ideally this should be no more than 20% of all utterances, since it directly questions whether the coding scheme actually measures the behaviors of interest. We only used "no code" in the conversation dimension. The number of utterances that we assigned this code was well below 20%. As discussed in the section on reliability, we did not include this code in the social and problem-solving dimensions as this would result in reliability overestimation due to sparse coding in these dimensions.

Holding Preliminary Try-outs

Calibration trials (or preliminary try-outs) should be based on a large enough number of observations in different groups, and/or different research conditions. In our case we made sure that each trial consisted of material from two different groups to prevent tuning the coding scheme to a single group. This practice makes the codes more universally applicable and improves reliability (consistency across different groups) and validity (identifying the same behavior in different groups). In general, several trials are required and about 10% of the data (depending on the frequency of behaviors and the number of codes) should be used in each trial to ensure that the sample is representative and behavior of interest actually occurs.

We conducted nine calibration trials to refine the set of codes constructed during the conceptual phase. During the first six trials the experienced content analysis researcher and two research assistants focused on the calibration of codes in the conversation and social dimension: adapting definitions, adding examples and adding rules to code ambiguous utterances. We discovered that conversational threading had to be reconstructed prior to coding the conversation dimension. In contrast to our conceptualization of the dimensions as being independent we had to allow 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. Usually a small amount of any given VMT chat falls into the social dimension, so in most chats utterances tied to problem-solving would also belong to the problem-solving dimension regardless of ties since most of the chat would be task-focused (i.e., solving the mathematical problem). Nevertheless, there will be instances where utterances in the social dimension are in fact technically of a more specific nature in a communicative sense than a mere "response" (this code was introduced to cover utterances not tied to problem-solving). The decision to allow for ties reflects our primary interest, that is, the mathematical problem solving. Nevertheless, we acknowledge that a stronger separation would have been preferred.

In trials seven to nine we focused on the problem-solving dimension and brought in three additional experts from the Math Forum team to assist with coding of mathematical problem solving. We concluded that a problem-solving thread had to be constructed prior to coding. An overview of possible solutions and strategies proved to be indispensable for coding problem solving. Yet, although we were able to identify problem-solving we had to concede that mathematical operations were too diverse and uncommon to achieve valid and reliable codes.

Developing Guidelines for Administration, Scoring and Interpretation of the Coding Scheme

In line with prior published coding schemes, we encountered ambiguous utterances that could be assigned several codes within a dimension. Ambiguous utterances are generally handled by establishing a set of rules. The number of rules should be limited as a need for many rules directly questions whether codes represent the behavior of interest (Beers et al., 2007). During the calibration trials we gradually accumulated rules to assist coding of ambiguous utterances. Two examples of rules for the conversation dimension are shown in Figure 22-1.

If an utterance is phrased as a question it is in general coded as a request. Sometimes a question mark is lacking, and it can be useful to use the preceding lines to determine the code. Exceptions:

- Although the use of a question mark may be guiding in assigning a "request," this can be misleading as occasionally utterances may be phrased as a question, when in fact they may be an "offer" in disguise, such as "**We need to calculate the height, right?**" In these cases the utterance is coded as an offer.
- If an utterance is framed as a question, but a specific responding conversational category applies to the content—often the content is a critique or regulate—the utterance is not coded as a request, but as critique or regulate.
- An utterance that consists only of a question mark is still coded as a "request" (? is a chat convention).

If the content of an utterance that has been coded as an "offer" or "elaborate" is phrased as a conclusion or the concluding step of a problem solving sequence, utterances following such an utterance—that contain "**Yes**"—are coded as agree. If the utterance that contains "**Yes**" is threaded to a solution step—which is not the final concluding step or utterance—this utterance is coded as "follow."

Figure 22-1. Sample rules for conversation codes.

We conducted two reliability trials. In each trial we used three coders (author and two research assistants). The first trial revealed an acceptable reliability for the conversation dimension, but the social and problem-solving dimensions needed to be refined and minor changes were made in the wording of a definition or adjusting a rule. The second trial revealed that the reliability for the social and problem-solving dimension improved, and reliability for the conversation dimension proved to be stable. An example of a coded transcript excerpt is shown in Log 22-2. (Compare qualitative analysis of the same log in Chapter 9.)

Log 22-2.

	Name	Text	Time	Delay	Ct	C	S	PSt	PS
32	AME	I have an idea that might help us find whats wrong with the pic.	06:19	00:49		s	is		
33	MCP	We could use good ol' Pythag thm to see what BV is	06:30	00:11		o	cg		s
34	AME	Lets not	06:40	00:10	33	d	cg	33	rf
35	MCP	What's your idea?	06:46	00:06	32	rq	ci	32	
36	AME	It states that something is wrong with the pic.	07:01	00:15	35	e		35	o
37	AME	so we can't find what BV is	07:08	00:07	36	el	cg	36	t
38	MCP	Yeah, and I think if we 'found' BV, it would be something not possible.	07:31	00:23	37	o	cg	37	t
39	MCP	16 + BV^2 = 21.16	08:10	00:39		o		33	p
40	MCP	BV^2 = 5.16	08:20	00:10	39	el		39	p
41	AME	I got it	08:23	00:03		se			
42	AME	I know whats wrong with the pic	08:29	00:06	41	s	is		
43	MCP	BV = 2.27	08:31	00:02	39	el		39	r
44	FIR	ok. now i'm following!	08:44	00:13	39	f	ci	39	

Note. Conversational thread (Ct), conversational dimension (C), social dimension (S), problem-solving thread (PSt) and problem-solving dimension (PS).

Empirical Evidence for Validity

In the end, the value of the coding scheme depends on whether the coding scheme is able to reveal the behaviors of interest. Empirical evidence for validity relates to Messick's (1989; 1995) consequential aspect of validity.

The purpose of the coding scheme was to describe collaborative processes of small groups solving a mathematical problem via chat. Once we had reliable coding of ten 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 saw a difference in interaction patterns that we termed expository versus exploratory (see Chapter 23).

Furthermore, the development of the VMT coding scheme and diversity of theoretical-methodological stances within the research team motivated the attempt to integrate the two seemingly disparate approaches: conversation analysis and coding. By using conversation analysis to construct a coding scheme—segmentation and codes based on the participants' view—statistical analyses revealed qualitative differences between chats in terms of activities that group members engaged in (e.g., socializing and problem solving), without violating the analytical requirements of either approach (see Chapter 23 again).

Finally, the VMT team investigated the expression and role of multiple voices in small-group chat communication (see Chapter 24). Evidence of multiple voices and differential social position with a corpus of chats could be expressed by the statistics of personal pronouns usage: "**I**" and "**me**" (appears in coding scheme as "collaboration individual") were used more often than "**we**" and "**us**" (appears as

"collaboration group" code); the second person addressing ("**you**") was well represented.

Nevertheless, even if analysis outcomes provide evidence that are deemed "valid", we should not forget that these outcomes are directly tied to what we "constructed" as an adequate representation of what might exist. Thus, however much our codes reflect a certain theory or perspective; we cannot assume that our representation fully covers the construct. At best a coding scheme reflects a more or less accurate approximation of what we intend to measure.

Generalizability

Regarding content analysis in collaborative learning research, Messick's (Messick, 1989; 1995) generalizability and external aspect are least addressed. Generalizability information is gathered through the re-use of a coding scheme in diverse contexts and knowledge domains, with diverse research populations and documenting whether similar behavioral patterns emerge.

Thus far, generalizability information has been accumulated for the Gunawardena *et al*. (1997) coding scheme (see De Wever et al., 2006), the Rainbow scheme (see Baker et al., 2007) and the Webb and Mastergeorge (2003) coding scheme (see Oortwijn et al., 2008). However, these examples account for a small fraction of coding schemes that have been developed and applied in collaborative learning research.

When judging generalizability information the source for variation should be kept in mind, i.e., different groups, different contexts and/or different domains. Furthermore, re-use of a coding scheme invariably leads to minor changes (e.g., adapting a definition, adding examples) or major changes (e.g., adding or deleting a code(s) or dimension)—tuning the coding scheme to the specific nature of the data collected or the research context (e.g., historical argumentation has features distinct from mathematical problem solving). The subsequent implications for reliability and validity should be addressed and carefully documented to foster re-use and accumulate validity evidence.

The external aspect has, thus far, only been addressed by Schellens & Valcke (2005), who coded the same data corpus with two coding schemes (Gunawardena et al., 1997; Veerman & Veldhuis-Diermanse, 2001) purportedly measuring the same construct. Irrespective of similarities there were differences as well, and there was evidence for convergent validity as "results confirm the theoretical mapping between phase 3 and 5 in the model of Veerman and phase 1 and phase 3 in the model of Gunawardena" (p. 972), but also divergent validity as other phases produced less similar results. In this respect it would be challenging—for example in the domain of argumentation in CSCL—to code argumentative knowledge construction in the same data corpus using both the Rainbow framework (Baker et al., 2007) and the Weinberger & Fischer (2006) framework.

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 multidimensional coding scheme to analyze interactions in these groups, four 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 posting 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 (the conversation-threading dimension) 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 larger group. Similarly to fragmentation, the connection between chat lines forming a chain of problem-solving 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 this response structure. Also, a threading coder must be familiar with the conversational codes. Assignment of both conversational and problem-solving threading connections is performed simultaneously and termed "threading." This represents 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 categorized as "no code" and included in the computation and (c) missing values and non-coded units are categorized 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 over-estimation. Including missing values as a "no code" is the strictest computation and a slight underestimation of the reliability. In our opinion a slight underestimation should be favored 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 e.g., 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. Conducting more than one reliability trial helped

to determine the impact of refinements (rewording definitions and changes to rules) and to assess reliability stability.

The fourth methodological issue concerns validity. 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. The VMT coding is explorative and draws on prior studies with content analysis, conversation analysis and ethnographic perspectives, which may have introduced some imbalance. Most codes are based on prior studies, but several codes emerged from working with the data. We spent considerable effort to establish the dimensions' independence, but were unable to achieve that. In principle this was due to codes such as explain, critique and elaborate that are historically connected to problem-solving rather than social issues. In reporting on an early stage in the VMT iterative, evolutionary design-based research of the VMT Project, we are not claiming that our coding scheme is the ultimate solution. It provided a starting point, based on our knowledge of existing coding schemes, some modification based on our research interests and on an inductive, grounded-research approach taken during the development and refinement of the scheme. We would certainly use a different set of codes now, based on our evolving understanding of the VMT student experience.

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, 2006b). 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 codes of actions generalized from past studies.

Finally, we are particularly interested in group cognition taking 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. 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 turned to conversation analysis to allow us 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 multidimensional coding of utterances could not capture the increasingly complex and innovative interactions (e.g., in Chapter 7). 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 codes, no matter how multidimensional 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, 2006a, Chapter 10). This is a paradox of reliable and valid coding efforts in exploratory CSCL research.

References

Andriessen, J., Baker, M., & Suthers, D. (Eds.). (2003). *Arguing to learn: Confronting cognitions in computer-supported collaborative learning environments*. Dordrecht, Netherlands: Kluwer Academic Publishers. Computer-supported collaborative learning book series, vol 1.

Artstein, R., & Poesio, M. (2005). *Kappa3 = alpha (or beta) (NLE technical note 05-1)*: University of Essex: Natural Language Engineering and Web Applications Group.

Baker, M., Andriessen, J., Lund, K., Van Amelsvoort, M., & Quignard, M. (2007). Rainbow: A framework for analysing computer-mediated pedagogical debates. *International Journal of Computer-Supported Collaborative Learning, 2*, 315-357.

Barron, B. (2003). When smart groups fail. *The Journal of the Learning Sciences, 12*(3), 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.

Beers, P. J., Boshuizen, H. P. A., Kirschner, P. A., & Gijselaers, W. H. (2007). The analysis of negotiation of common ground in CSCL. *Learning & Instruction, 17*, 427-435.

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.

Cress, U. (2008). The need for considering multilevel analysis in CSCL research: An appeal for more advanced statistical methods. *International Journal of Computer-Supported Collaborative Learning, 3*, 69-84.

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. Special issue on measurement challenges in collaborative learning research, 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.

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*, 397-343.

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

Hmelo-Silver, C. (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.
Messick, S. (1989). Meaning and values in test validation: The science and ethics of assessment. *Educational Researcher, 18*(2), 5-11.
Messick, S. (1995). Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning. *American Psychologist, 50*, 741-749.
Neuendorf, K. A. (2002). *The content analysis guidebook*. Thousand Oaks, CA: Sage.
Oortwijn, M. B., Boekaerts, M., Vedder, P., & Strijbos, J. W. (2008). Helping behaviour during cooperative learning and learning gains: The role of the teacher and of pupils' prior knowledge and ethnic background. *Learning & Instruction, 18*, 146-159.
Polya, G. (1945/1973). *How to solve it: A new aspect of mathematical method*. Princeton, NJ: 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. (2004). Validity in quantitative content analysis. *Educational Technology Research & Development, 52*, 5-18.
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. *The Journal of the Learning Sciences, 11*(2 & 3).
Stahl, G. (2006a). *Group cognition: Computer support for building collaborative knowledge*. Cambridge, MA: MIT Press. Retrieved from http://GerryStahl.net/mit/.
Stahl, G. (2006b). Sustaining group cognition in a math chat environment. *Research and Practice in Technology Enhanced Learning (RPTEL), 1*(2), 85-113. Retrieved from http://GerryStahl.net/pub/rptel.pdf.
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. Retrieved from http://GerryStahl.net/cscl/CSCL_English.pdf in English, http://GerryStahl.net/cscl/CSCL_Chinese_simplified.pdf in simplified Chinese, http://GerryStahl.net/cscl/CSCL_Chinese_traditional.pdf in traditional Chinese,

http://GerryStahl.net/cscl/CSCL_Spanish.pdf in Spanish,
http://GerryStahl.net/cscl/CSCL_Portuguese.pdf in Portuguese,
http://GerryStahl.net/cscl/CSCL_German.pdf in German,
http://GerryStahl.net/cscl/CSCL_Romanian.pdf in Romanian.

Strijbos, J. W. (2004). *The effect of roles on computer supported collaborative learning.* Unpublished Dissertation, Ph. D., Open Universiteit Nederland, Heerlen, the Netherlands.

Strijbos, J. W., & Fischer, F. (2007). Methodological challenges for collaborative learning research. *Learning & Instruction, 17*, 389-393.

Strijbos, J. W., Kirschner, P. A., & Martens, R. L. (Eds.). (2004). *What we know about CSCL ... And implementing it in higher education*. Dordrecht, the Netherlands: Kluwer Academic Publishers. Computer-supported collaborative learning book series, vol 3.

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.

Suthers, D. (2005). Technology affordances for intersubjective learning, and how they may be exploited. In R. Bromme, F. W. Hesse & H. Spada (Eds.), *Biases and barriers in computer-mediated knowledge communication: And how they may be overcome* (pp. 295-319). Boston, MA: Kluwer Academic Publishers.

Veerman, A., & Veldhuis-Diermanse, E. (2001). Collaborative learning through computer-mediated communication in academic education. In P. Dillenbourg, A. Eurelings & K. Hakkarainen (Eds.), *European perspectives on computer-supported collaborative learning* (pp. 625-632). Maastricht, the Netherlands: University of Maastricht.

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.