

The analysis of negotiation of common ground in CSCL

Citation for published version (APA):

Beers, P., Boshuizen, E., Kirschner, P. A., & Gijssels, W. (2007). The analysis of negotiation of common ground in CSCL. *Learning and Instruction*, 17(4), 427-435. <https://doi.org/10.1016/j.learninstruc.2007.04.002>

DOI:

[10.1016/j.learninstruc.2007.04.002](https://doi.org/10.1016/j.learninstruc.2007.04.002)

Document status and date:

Published: 01/08/2007

Document Version:

Peer reviewed version

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

Take down policy

If you believe that this document breaches copyright please contact us at:

pure-support@ou.nl

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 04 Jul. 2022

Open Universiteit
www.ou.nl



Running head: ANALYSIS OF NEGOTIATION IN CSCL

This article was published as:

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

Copyright Elsevier, available online at

http://www.elsevier.com/wps/find/journaldescription.cws_home/956/description#description

The Analysis of Negotiation of Common Ground in CSCL

Pieter J. Beers,

Delft University of Technology / Open University of the Netherlands

Henny P. A. Boshuizen

Open University of the Netherlands

Paul A. Kirschner

Utrecht University / Open University of the Netherlands

Wim H. Gijsselaers

Maastricht University

Author Note

Correspondence concerning this article should be addressed to Pieter J. Beers, Delft University of Technology, Faculty of Technology, Policy and Management, P.O. Box 5015, 2600 GA Delft, The Netherlands. E-mail may be sent to p.j.beers@tudelft.nl

Abstract

CSCL research has given rise to a plethora of analysis methods, all with specific analysis goals, units of analysis, and for specific types of data (chat, threaded discussions, etcetera). This article describes some challenges of CSCL-analysis. The development of an analysis method for negotiation processes in multidisciplinary teams serves as an example of how these challenges occur in practice. Results reveal a number of tensions with regard to establishing reliable coding procedures without compromising validity, such as how to define codes, and rules for applying codes. The discussion offers some guidelines for content analysis of CSCL-data.

1. The Analysis of Negotiation of Common Ground in CSCL

The recent growth in CSCL research has given rise to a plethora of methods for the analysis of communication processes. Well-known examples are content coding schemes for the analysis of collaborative construction of knowledge (Gunawardena, Lowe, & Anderson, 1998), learning process in computer conferencing (Henri, 1992), and co-construction of understanding and knowledge (Zhu, 1996). However, most such methods are poorly documented, even though their development is often fraught with numerous choices and problems. Every CSCL analysis method uses its own specific unit of analysis, has its own specific analysis goals, and is made for a specific data type (chat, threaded discussions, etcetera). Also, in the course of their development, CSCL-environments are often used in various educational and research settings, which may differ with respect to things like synchronicity, task length (from an afternoon to a couple of months), task interdependence (mutual learners' dependences for executing the task). This variety in both goals and settings can turn the analysis of CSCL-effects into a daunting task.

When studying effects of CSCL, researchers tend to use measures of the communication process (content analysis) to gain insight in the way learning takes place (Chi, 1997; Strijbos, Martens, Prins, & Jochems, 2006). The difficulty with doing content analyses of communication stems from a lack of guidelines for performing them validly and reliably (Rourke & Anderson, 2004; Strijbos et al., 2006). This is to say that current know-how of content analysis remains largely undescribed and implicit, as scientific publications generally are too short to fully explain the procedures employed and the problems encountered during analysis. The main goal of this article is to make some of these procedures explicit.

The challenge of analysis in CSCL concerns developing content analysis methods that are both structured enough to achieve reliability and validity, and that can be used in various experimental and educational contexts. These issues have received quite some attention in recent literature. For example, Rourke, Anderson, Garrison, and Archer (2001) presented a set

of guidelines to achieve validity by identifying fundamental issues such as objectivity, reliability, replicability, and systematic consistency. Furthermore, Strijbos (2004) showed how different treatments of segmenting CSCL-data can influence reliability by radically separating the segmentation procedure from the actual coding of the data. Nonetheless, little is known about the actual validity and reliability problems that emerge in practice throughout the development of a new analysis coding scheme. Developing content analysis methods is further complicated because CSCL-development often covers the full range of face-to-face prototypes to practical (distributed, asynchronous) implementation (Kirschner, Strijbos, Kreijns & Beers, 2004; Nieveen, 1999).

This article reports on the development of, and experiences with content analysis as part of a research project about “Knowledge sharing and knowledge building in expert teams with ICT.” The project involves developing ICT-support for negotiation of common ground. The associated analysis problem was how to qualitatively measure negotiation of common ground in communication processes in both face-to-face situations and computer-mediated threaded discussions in a quantifiable manner, and doing this both reliably and validly.

The purpose of this article is to describe a practical example of content-analysis, and to draw some general guidelines for such analyses. First we describe some challenges of content analysis regarding validity and reliability in more detail. Then we discuss the present research context and give a short overview of its theoretical background. The article proceeds with a discussion of issues dealt with during the development of one specific coding scheme. Finally, some guidelines for CSCL-analysis are presented in the Discussion.

2. Validity and Reliability in Content Analysis

CSCL-environments generally use specific rules and constraints for communication (a *formalism*), either as a result of planning on the part of the developers (i.e., we want certain communication), or as an artefact of the system or software used (e.g., a text-based system does not allow for diagrams or flowcharts). A formalism can be seen as a set of rules and

objects that together model a type or aspect of conversation (Dillenbourg, 2002). A formalism is often based on educational theories that describe how it can be used to solve a learning problem. For instance, different formalisms have been developed to support such diverse processes as group design (Buckingham Shum, MacLean, Bellotti & Hammond, 1997), scientific reasoning (Suthers, 2001), and argumentation (Van Bruggen, 2003). Buckingham Shum et al. designed a formalism called the QOC (Questions, Options, Criteria) approach to group design. With QOC, design problems are structured in terms of design *questions*, *options* for answering the questions, and *criteria* to be fulfilled in order for the options answer the questions. Rules are that for each option, it is assessed to what extent it answers the question design, using the criteria.

Each specific educational goal requires an associated specific analysis method, as the intended effects of the formalisms will influence communication in specific ways. This also means that existing analysis methods cannot be used for analysing data from new CSCL-environments. In other words, new CSCL development-projects will generally require new coding schemes for analysis.

Several researchers (e.g. Chi, 1997; Rourke & Anderson, 2004) have described which steps they took in developing a CSCL *coding scheme* (see Table 1). Development starts with identifying the main analysis goal and defining codes that validly represent the construct under analysis. The next step involves identifying those communication aspects that typify each code, including segmentation of data. Segmentation is important because the grain-size of data segments needs to be at the same level at which the typifying communication aspects occur (Chi, 1997). Some researchers completely separate segmentation from coding (Strijbos et al., 2006) whereas others use the code definitions only and leave it to the coders to identify appropriately sized segments. The third step is defining rules for coding so that the coding scheme can be used reliably. In our case the analysis goal is negotiation of common ground.

With regard to segmentation, if one would, for example, want to code instances of successful negotiation episodes, s/he cannot take the sentence or a 'turn' as a segment, because successful negotiation episodes consist of a number of conversation turns taken by different participants. The sentence and the turn level are too fine-grained to identify negotiation episodes.

Table 1 about here

The codes and their identifiers being identified, the next step is making *rules* for the actual administration of the codes so as to be able to use the coding scheme reliably. Although theoretically a segmented data set with a description of the codes and identifiers should result in a reliable coding scheme, the reality of CSCL analysis is often different.

Achieving reliability greatly depends on the type of construct that is being coded (Rourke et al., 2001). If the construct concerns manifest content, which can be coded with little need for subjective interpretation, achieving high reliability is relatively easy. On the other hand, coding latent content, which cannot be read from surface aspects of the data and which requires subjective interpretation, will require elaborate guidelines for scoring and interpretation. In other words, the extent to which data are subjectively ambiguous with respect to the codes influences the reliability of the coding scheme. Some authors argue for some lenience as to a criterion for 'good' reliability, especially in the case of coding highly latent content, because aiming for reliability can sometimes hinder validity (Krippendorff, 1980).

A coding scheme can be considered complete when it validly represents the construct of interest, and when coders can reliably apply the codes in practice. The challenge for the researcher thus concerns the development of such rules for the administration of codes such that they lead to reliable results, without compromising the original meanings of the code definitions.

3. A Coding Scheme for the Analysis of Negotiation of Common Ground

In the case of our own studies, we did not know of, nor could we find any valid coding schemes for the analysis of negotiation of common ground, thus we needed to develop a new one. In the following sections we will describe how we dealt with issues of reliability and validity throughout developing our own coding scheme. However, first we will describe the project context in which the analysis scheme was applied. The coding scheme was developed in the context of the design of an ICT-tool to support the negotiation of common ground in multidisciplinary teams, NegotiationTool (NTool).

The project set out from the notion that multidisciplinary can be an asset for complex problem solving (Lomi, Larsen, & Ginsberg, 1997; Nutt, 2002), because it is thought that solving complex problems requires integration of knowledge from different disciplines (Rotmans, 1998). However, to benefit from multidisciplinary, the team members need to be able to effectively share knowledge with each other and understand each other. In other words, those teams need enough common ground, a shared cognitive frame of reference (Bromme, 2000), for their purposes of collaboration. Common ground is especially important for multidisciplinary teams, as they tend to have little common ground to begin with (Alpay, Giboin, & Dieng, 1998). The researchers used theory about the negotiation of common ground as a basis for the design of a formalism to support multidisciplinary teams (Beers, Boshuizen, Kirschner, & Gijsselaers, 2005).

Theory on negotiation of common ground originated in linguistics (Clark & Schaefer, 1989; Clark & Brennan, 1991) and cognitivism (Bromme, 2000). The linguistic approach describes how negotiation of common ground appears in conversation, whereas the cognitivist approach focuses on the way new knowledge is processed, the role that prior knowledge plays in this process, and how individual perspectives affect this. The combination of the two links the content of a learning process to the way it is communicated between people. Based on these theories, the grounding process starts when team members contribute their, as yet,

unshared knowledge, so that others can try to apprehend that knowledge. At this point, a number of biases come into play that cause differences between the intended meaning of a contribution, and the contribution as it is understood. While constructing their own individual understanding, the other team members use their knowledge of aspects like the contributor's background and views held, and the current situation, to better "understand" the contribution. Also, their own beliefs and assumptions affect their understanding of a contribution. A contribution is thus always understood against the *presumed perspective of the other* and *one's own perspective* (Bromme, 2000), and the negotiation of common ground then is the iterative minimisation of these representational differences, through providing feedback based on one's own perspective by word or action (Alpay et al., 1998; Baker et al., 1999).

As mentioned above, a formalism can be seen as a set of rules and objects that together model conversation (Dillenbourg, 2002), with the aim to enhance a group process. The definition of a formalism thus requires the translation of a theory about a phenomenon to a theory of how that phenomenon *should* occur in actual communication. In the case of negotiation of common ground, the step from a theoretical framework of negotiation of common ground to a formalism for the support of negotiation is fairly straightforward, as theory of negotiation originated in linguistics (Clark & Schaefer, 1989) and cognitivism (Bromme, 2000); from Bromme's notions of one's own perspective and the presumed perspective of the other, it was deduced that the negotiation of common ground could be supported by encouraging team members to explicitly verify their understanding (and clarify upon misunderstanding) of each others' contributions, so as to minimise mismatches due to wrongly presuming another's perspective, and due to reasoning and communicating from one's own perspective only. This was used as the basis for a *prescriptive* set of communication rules (the formalism, see Table 2) that encouraged team members to mutually verify and clarify their understanding of others' contributions during discussion, to explicitly accept or reject contributions, and to make their positions known to their team-mates.

Table 2 about here

Note that a formalism thus can be very different from a *descriptive* framework, such as the one by Clark and Schaefer (1989), which describes how the *actual* grounding process unfolds. These differences will be made clearer in the following sections. Also note that the formalism is particularly task-oriented, that is, it aims to facilitate grounding activities related to the task at hand and its informational content, and disregards other grounding activities. In that sense, the formalism theoretically departs from both Clark and Schaefer (1989) and Bromme (2002).

The formalism was first used in two face-to-face experiments, in which participants received written instructions that embedded the formalism rules for the use of a set of writing materials (markers, whiteboard, and flipover). Next, the formalism was embedded in NTool and used in two other studies. NTool is a newsclient-based computer-mediated communication tool that uses text messages with a small predefined set of message types to facilitate the negotiation process. In all studies, be it face-to-face or with NTool, multidisciplinary teams collaboratively solved a complex problem. The problems were chosen so that all team members were able to apply their own disciplinary knowledge to them. For instance, in two of our studies groups of three students (one student of economics, one of cultural sciences, and one of psychology in each group) were assigned the task to collaboratively solve the complex problem of school drop-out. Negotiation was analysed through study of the communication process using content analysis techniques to create a process overview of negotiating common ground. In the case of the face-to-face studies, the communication was videotaped and analysed with video-coding software (The Observer®; Noldus, Trienes, Hendriksen, Jansen, & Jansen, 2000). In the case of the computer-mediated discussions the communication was cached on a server and accessible with NTool.

All reliability measures were computed based on samples of the whole data set, coded by the first author of this paper and an assistant. After reliability was established, the whole body of data was coded by the assistant alone.

4. Validity and Reliability

4.1. Definition of the Codes

The analysis goal was to identify those aspects of communication that dealt with the negotiation of common ground. To that end we used the prescriptive account of the grounding process (the formalism) for our initial code definitions. The main advantage of this approach is that it quickly bridges the difference between theory and codes, in the same way as the formalism does. This strategy, using the same theory for both the design of CSCL and the analysis of its effects, is also described by Chi (1997).

The above procedure resulted in the following codes for cognitive conversation content (directly related to solving the problem):

- Contribution: A new topic of conversation that has not been discussed before is introduced.
- Verification: Information is directly or indirectly requested about the intended meaning of a contribution or elaboration.
- Clarification: A reaction to a verification or a perceived lack of understanding, in which the intended meaning of a contribution or elaboration is elucidated.
- Acceptance: A reaction to a contribution in which the contribution is judged intelligible and/or correct.
- Rejection: A reaction to a contribution in which the contribution is judged unintelligible and/or incorrect.
- Agreement: A reaction to a contribution in which the sender voices his/her agreement with the contribution.

- Disagreement: A reaction to a contribution in which the sender voices his/her disagreement with the contribution.

Note that these subcategories are actually descriptions of the *function* of a turn in a part of conversation that deals with task *content*. This combination of content-coding and functional coding has been detailed by Fischer et al. (2002).

Also note that this use of the term “contribution” differs from the original use by Clark and Schaefer (1989) in that their use presumes that a contribution constitutes a number of communicative moves by two or more discussion partners that effectively makes something said part of common ground. In their (linguistic) view, a contribution typically takes about 13 words, although it may include some breakdowns of understanding and subsequent repair. In contrast, in the above set of codes the word “contribution” refers only to the conversation turn in which *one* individual starts to share a new idea with his/her team-mates, with breakdowns and repairs of understanding being covered by the other codes. Furthermore, in our studies the number of words per contribution greatly exceeded 13, as they generally consisted of a number of sentences. Our coupling communicative moves with task content may have contributed to this difference; discussing one contribution in terms of task content may involve a number of contributions in linguistic terms.

4.2. *Identification of Communicative Identifiers and Segmentation*

Using the above code definitions for cognitive conversation content made it possible to identify negotiation. However, in many cases the communication did not fit any of the above subcategories. This necessitated the development of further codes. First of all, people building on each other (cognitive content) without explicitly negotiating common ground was coded *Elaboration*: A contribution is elaborated upon by adding information or summarising.

An example of an elaboration and the associated judgements to be made is the following: “That’s a good idea, but of course there are exceptions.” In the example there is no apparent questioning of one’s own understanding (i.e., it is not a verification) or clarifying of

an original contribution (i.e., it is not a clarification), there is no explicit agreeing or disagreeing (of course “good idea” does show a positive attitude, but “exceptions” shows that there are conditions to be met before the speaker can wholly agree), or explicit acceptance or rejection. However, it still represents task content, and furthermore, the “exceptions” mentioned add information to the discussion that was not mentioned before without introducing a new conversation topic which leaves us only the *Elaboration* category.

Note that our coding again departs from Clark and Schaefer’s (1989) process description. In their conceptualisation, “*That’s a good idea, . . .*” refers to previous conversation content that, if not disputed, belongs to common ground.

Furthermore, cognitive content communication was distinguished from *task regulation* and *other* communication. These categories were not in our original theoretical framework, but came from experiences while applying the content codes. It occurred to the coders that large amounts of communication were neglected with the original codes, and that these roughly covered conversation with the aim of regulating the task and ‘other’ communication. This procedure coincides with an open-coding approach to qualitative research (see, e.g., Marton, 1981; Strauss, 1987). Two regulative categories were used; messages with regulative content that addressed monitoring the problem solving-process (e.g., “I think we have not yet arrived at a good problem definition;” coded Monitoring), and messages that only regulated the conversation (e.g., “Could you make a note of that on the board please;” coded Regulation, also including the regulation of using the NTool, where appropriate).

The segmentation procedure was not separated from the actual coding. The code definitions indicated the use of the conversation turn (in case of face-to-face communication) or the message (in case of computer-mediated communication). A conversation turn or an NTool message would only be split in two segments when it contained a clear change in terms of content. The resulting segments were not as fine-grained as the examples of grounding

described by Clark and Schaefer (1989), which indicates a difference between Clark and Schaefer's (1989) descriptive accounts of negotiation and our coding scheme.

4.3. *Rules for Administration*

Using only the codes derived from theory was not reliable; additional rules for coding were needed to deal with ambiguous communication. We used two different strategies to arrive at acceptable reliability, (1) using a set of decision rules for ambiguous cases, and (2) recoding several categories into one. The rules for coding "Yes," and "No," in face-to-face communication are a good example of the first strategy (see Table 3).

The main problem of the meaning of "Yes," and "No," lies in the fact that they can be used in so many different ways, with so many different meanings. A rough approximation in writing of this phenomenon would be the difference between "Yes!" and "Yes. . ." Especially in the case of the codes agreement, disagreement, acceptance, and rejection, the words "Yes," and "No" were important sources of disagreement between coders, because the actual meaning is very much context-dependent. For instance, "Yes," is an agreement if following a question like "Would you agree that additional teachers can mitigate the problem of school drop-out?" whereas it is acceptance following a question like "So am I correct that you think additional teachers will mitigate the problem of school drop-out?" Table 3 shows that four of the additional rules describe contextual conditions for coding "Yes," and "No." We decided to circumvent this issue by only coding entirely explicit agreements, disagreements, acceptances, and rejections, and using a specific set of rules for coding "Yes," and "No."

Table 3 about here

Unfortunately, this resulted in very low numbers of agreement, disagreement, acceptance and rejection. This means that the actual numbers coded were lower than would have been justified on the basis of the code definitions alone. This is a clear example of the strain that can occur between validity and reliability.

Also, the above strategy alone did not result in good reliability. A second, somewhat more radical measure was recoding several categories into one. In the first study, codes for acceptance, rejection, agreement, and disagreement were recoded as elaboration, which resulted in an inter-coder reliability (Cohen's kappa) of .68. The analysis of the second face-to-face experiment, with a different assistant, included codes for agreement and acceptance (kappa = .71), but not for rejection and disagreements because the number of disagreements and rejections was too low for computing reliability statistics.

The analysis scheme was also used twice for coding computer-mediated communication, with an inter-coder reliability of .70, including all content codes and rules used earlier, except segmentation. Coding may have been easier for electronic communication since electronic communication is naturally segmented into a unit (the message) that is meaningful for the coding scheme.

In sum, defining the initial codes did not pose any specific problems, but our focus on task content has led us to depart to some extent from Clark and Schaefer's (1989) original account of the negotiation process. Also, we needed to develop more codes because large amounts of data remained uncoded with only our initial set of codes. To be able to code reliably, we included some additional rules for cases of doubt, and recoded some codes into others.

5. Discussion

This article aimed to describe a practical example of content-analysis, and to draw some general guidelines for such analyses. Here we summarise our experiences throughout the development of our coding scheme and draw some guidelines from them. Literature emphasises the tensions between validity and reliability, especially in the case of coding latent variables. We developed a coding scheme using a three-step-procedure, (1) defining the codes, (2) identifying code-specific communication and segmentations, and (3) making rules to apply the coding scheme reliably.

In our experience, the first step – defining the codes – was fairly straightforward because the specific theories we used already contained some information about their occurrence in communication, being partly derived from linguistics. Therefore, it was easy to translate between theory and actual communication using the same theory for both the formalism and the coding scheme, as proposed by Chi (1997). However, codes theoretically departed from Clark and Schaefer's (1989) descriptions of the grounding process. This was the consequence of our use of a *prescriptive* theoretical framework instead of a *descriptive* one, and of our interest in task content instead of communicative moves, and we adapted our code definitions and procedures accordingly. The point to be made is that neither Clark & Schaefer (1989) nor Bromme (2000) describe the characteristics of how a good grounding process should occur in communicative terms, which begs the question of whether to code the communication process as it was originally proposed by Clark and Schaefer (1989), or to code the communication process as conceived by the formalism. We opted for the latter in our studies.

For defining the codes, we propose the following as a guideline. When researchers aim to test differences between CSCL-environments, for instance between different versions of a formalisms, or with and without a formalism, we propose to define codes based on the formalism, like we did here. That means, to aim for theoretical consistency with the formalism, but not necessarily the underlying descriptive theories. That way, one is able to use a small set of codes tailored to a specific purpose, which is important for keeping the labour-intensiveness of coding within limits, especially because testing statistical differences demands a large number of code counts to have enough statistical power to detect differences.

The case for an exploratory study of how a specific process occurs in CSCL is different, as this does not necessarily demand coding the entire body of data (less labour-intensive, no disadvantages of having to code a huge number of very fine-grained segments). In those cases we propose that it is best to use existing descriptive theories (if any exist) of the

communication process. Obviously, an open coding approach (no predefined codes) is needed when no previous theory exists.

The second step – identification of code-specific communication characteristics and segmentation – showed that our initial code definitions were a good starting point, but also that a lot of the communication did not fit any of our predefined categories. Developing new codes for a number of cases enabled coding all communication. We did not encounter specific segmentation problems. Apparently, the conversation turn as a segmentation criterion sufficiently fit the grain-size of the codes for it to be unproblematic.

With regard to developing codes, it is important to note that using only codes based on a formalism will probably lead to a large amount of uncoded data. Researchers should be prepared to add new codes to decrease the amount of non-coded data. This may enable coding more reliably, because it can be easier to code something for what it is than to hesitate about whether or not to use one of the original codes. Also, such codes may later help in explaining unexpected results. For example, in our own studies adding codes for regulation helped us to explain some results that turned out to be related to one specific experimental setting (see Beers et al, 2005).

With regard to segmentation, we propose that researchers should prevent separating coding and segmentation when they are able to do so reliably, but keep in mind that the data segments produced this way may lead to a process description that differs significantly from theoretical accounts.

The third step – the development of rules for the administration of codes – was indispensable for achieving reliability, but it also endangered validity. The main problem with reliability and validity probably stemmed from the very latent character of the codes, which means that data were likely to be ambiguous for the coders, even if the coders understood and agreed about the definitions of the codes. Nonetheless, the ultimate validity and reliability of

the resulting coding schemes and rules for administration were sufficient to draw conclusions with regard to the research questions within the project.

We advise to either eliminate any codes that occur rarely or not at all from the analysis, or to broaden their definition so that it occurs often enough for statistical analysis. Obviously, this may again pose reliability problems (out of the frying pan, into the fire). In those cases where reliability can still not be achieved, discussing cases of dissent between coders to establish intersubjectivity can be a solution. For instance, one might compute a reliability statistic before discussion, identify cases of dissent, and discuss them to reach agreement, and report the reliability figure again. In some cases, this may iron out some misapplication of certain codes, or differences in understanding of the codes between the coders. However, it may also mean that the coders have different interpretations of the data itself. The difficulties with coding “Yes,” and “No,” are an example in case. Different coders may have the same understanding of what the code definition of agreement is, but still have very different perceptions of whether or not a specific statement indeed *is* an agreement.

A general point needs to be made about the use of coding schemes developed by others. Earlier in this article we claimed that new CSCL-development projects generally require new analysis schemes. Based on the above, it could be argued that using rules for administration developed by others as well as their code definitions can enable the use of others’ coding schemes, that is, in the rare case where an existing coding scheme is applicable. However, rules for administration of codes are generally absent from scientific articles that employ content analysis (Strijbos et al., 2006). Furthermore, the present results show that each coder has his/her own capacities of understanding a coding scheme, and that individual differences may require different rules for the administration of codes. Thus, the availability of others’ rules for the administration might increase the adoptability of others’ coding schemes, but it is not a guarantee for reliability.

In general, CSCL-analysis must take both validity of the coding scheme and the associated measurement strategy into account. Furthermore, it is important to distinguish between these two aspects of analysis, as the one may compromise the other. Aiming for reliability can render a coding scheme invalid if one applies a measurement strategy without proper care for validity.

Acknowledgements

We would like to thank our colleagues at OTEC, and our team members Mien Segers, Piet Van den Bossche and Emily Teunissen for their input and ideas. This research was funded by the Dutch National Organisation for Scientific Research (NWO).

References

- Alpay, L., Giboin, A., & Dieng, R. (1998). Accidentology: An example of problem solving by multiple agents with multiple representations. In M. W. Van Someren, P. Reimann, H. P. A. Boshuizen & T. De Jong (Eds.), *Learning with multiple representations* (pp. 152-174). Oxford, UK: Elsevier.
- Beers, P. J., Boshuizen, H. P. A., Kirschner, P. A., & Gijssels, W. H. (2005). Computer support for knowledge construction in collaborative learning environments. *Computers in Human Behavior*, 21(4), 623-643.
- Bromme, R. (2000). Beyond one's own perspective: The psychology of cognitive interdisciplinarity. In P. Weingart & N. Stehr (Eds.), *Practicing interdisciplinarity* (pp. 115-133). Toronto, Canada: University of Toronto Press.
- Buckingham Shum, S. J., MacLean, A., Bellotti, V. M. E., & Hammond, N. V. (1997). Graphical argumentation and design cognition. *Human-Computer Interaction*, 12, 267-300.
- Chi, M. T. H. (1997). Quantifying qualitative analyses of verbal data: A practical guide. *The Journal of the Learning Sciences*, 6(3), 271-315.
- Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnick, J. M. Levine & S. D. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 127-149). Washington DC, USA: American Psychological Association.
- Clark, H. H., & Schaefer, E. F. (1989). Contributing to discourse. *Cognitive Science*, 13, 259-294.
- Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL: Can we support CSCL?* (pp. 61-91). Heerlen, The Netherlands: Open Universiteit Nederland.
- Fischer, F., Bruhn, J., Gräsel, C., & Mandl, H. (2002). Fostering collaborative knowledge construction with visualization tools. *Learning and Instruction*, 12, 213-232.

- Gunawardena, C. N., Lowe, C. A., & Anderson, T. (1998, August). Transcript analysis of computer-mediated conferences as a tool for testing constructivist and social-constructivist learning theories. Paper presented at the Fourteenth Annual Conference on Distance Teaching and Learning, Madison, WI.
- Henri, F. (1992). Computer conferencing and content analysis. In A. R. Kaye (Ed.), *Collaborative learning through computer conferencing: The najaden papers* (pp. 117-136). Berlin: Springer.
- Kirschner, P., Strijbos, J.-W., Kreijns, K., & Beers, P. J. (2004). Designing electronic collaborative learning environments. *Educational Technology Research and Development*, 52(3), 47-66.
- Krippendorff, K. (1980). *Content analysis: An introduction to its methodology*. Beverly Hills, US: Sage.
- Lomi, A., Larsen, E. R., & Ginsberg, A. (1997). Adaptive learning in organizations: A system-dynamics-based exploration. *Journal of Management*, 23(4), 561-582.
- Marton, F. (1981). Phenomenography - describing conceptions of the world around us. *Instructional Science*, 10, 177-200.
- Nieveen, N. (1999). Prototyping to reach product quality. In J. van den Akker, R. M. Branch, K. Gustafson, N. Nieveen & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 125-135). Dordrecht, the Netherlands: Kluwer Academic Publishers.
- Noldus, L. P. J. J., Trienes, R. J. H., Hendriksen, A. H. M., Jansen, H., & Jansen, R. G. (2000). The Observer Video-Pro: New software for the collection, management, and presentation of time-structured data from videotapes and digital media files. *Behavior Research Methods, Instruments & Computers*, 32, 197-206.
- Nutt, P. C. (2002). *Why decisions fail*. San Francisco: Berrett-Koehler.

- Rotmans, J. (1998). Methods for IA: The challenges and opportunities ahead. *Environmental Modeling and Assessment*, 3(3, Special Issue: Challenges and opportunities for Integrated Assessment), 155-179.
- Rourke, L., & Anderson, T. (2004). Validity in qualitative content analysis. *Educational Technology Research and Development*, 52(1), 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.
- Strijbos, J.-W. (2004). *The effect of roles on computer-supported collaborative learning*. Unpublished PhD-thesis, Open University of the Netherlands, Heerlen, the Netherlands.
- 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.
- Strauss, A. L. (1987). *Qualitative analysis for social scientists*. Cambridge, UK: Cambridge University Press.
- Suthers, D. D. (2001). Towards a systematic study of representational guidance for collaborative learning discourse. *Journal of Universal Computer Science*, 7(3). Retrieved 23 January 2004, from http://www.jucs.org/jucs_7_3/towards_a_systematic_study
- Van Bruggen, J. M. (2003). *Explorations in graphical argumentation; The use of external representations in collaborative problem solving*. Unpublished PhD-thesis, Open University of the Netherlands, Heerlen, The Netherlands.
- Zhu, E. (1996). *Meaning negotiation, knowledge construction, and mentoring in a distance learning course*. Paper presented at the National Convention of the Association for Educational Communications and Technology.

Table 1

Steps for the Development of a Coding Scheme

-
1. Definition of the codes
 2. Identification of code-specific communication characteristics and segmentation
 3. Development of coding rules to achieve reliability
-

Table 2

NTool Formalism

Every new issue is termed a *contribution*

Contributions require *verification* by the other team members

Verification is responded to with *clarification* by the original contributor

When all verifications are clarified, and no new verifications are performed, all

team members state their *acceptance* or *rejection* of the statement

All team members state their *position* about accepted statements

Table 3

Rules for Reliably Coding “Yes” and “No”

-
1. Clarification, when “Yes” or “No” immediately follows a Verification in the form of a closed question
 2. Acceptance or Rejection, when “Yes” or “No” immediately follows a Clarification or Elaboration about the meaning of something that has not been part of a Contribution before, and in which confirmation is asked.
 3. Agreement or Disagreement, when “Yes” or “No” is the answer to a question about another’s opinion.
 4. Elaboration, if in any of the above cases “Yes” or “No” is followed by a conditional statement.
 5. Elaboration when in doubt.
-