Computer-Supported Collaborative Learning
DOI 10.1007/s11412-008-9052-6

Automatic coding of dialogue acts in collaboration protocols

Gijsbert Erkens · Jeroen Janssen

Received: 17 May 2007 / Accepted: 18 August 2008

Abstract Although protocol analysis can be an important tool for researchers to investigate the process of collaboration and communication, the use of this method of analysis can be time consuming. Hence, an automatic coding procedure for coding dialogue acts was developed. This procedure helps to determine the communicative function of messages in online discussions by recognizing discourse markers and cue phrases in the utterances. Five main communicative functions are distinguished: *argumentative*, *responsive*, *informative*, *elicitative*, and *imperative*. A total of 29 different dialogue acts are specified and recognized automatically in collaboration protocols. The reliability of the automatic coding procedure was determined by comparing automatically coded dialogue acts to hand-coded dialogue acts by a human rater. The validity of the automatic coding procedure was examined using three different types of analyses. First, an examination of group differences was used (dialogue acts used by female versus male students). Ideally, the coding procedure should be able to distinguish between groups who are likely to communicate differently. Second, to examine the validity of the automatic coding procedure through examination of experimental intervention, the results of the automatic coding procedure of students, with access to a tool that visualizes the degree of participation of each student, were compared to students who did not have access to this tool. Finally, the validity of the automatic coding procedure of dialogue acts was examined using correlation analyses. Results of the automatic coding procedure of dialogue acts of utterances (form) were related to results of a manual coding procedure of the collaborative activities to which the utterances refer (content). The analyses presented in this paper indicate promising results concerning the reliability and validity of the automatic coding procedure for dialogue acts. However, limitations of the procedure were also found and discussed.

Keywords Collaborative learning · Computer-supported collaborative learning · Protocol analysis · Dialogue acts

G. Erkens (✉) · J. Janssen
Research Centre Learning in Interaction, Utrecht University, P.O. Box 80140, Utrecht, The Netherlands
e-mail: G.Erkens@fss.uu.nl

Introduction

Researchers seem to agree that the interaction between group members is the mechanism which fosters students' learning during collaborative learning, whether online or face-to-face (cf., De Wever et al. 2006; Kreijns et al. 2003). During computer-supported collaborative learning (CSCL), the interaction between group members is recorded in protocols of the online collaboration process. The study of these protocols has been the focus of much research. Research on the process of collaboration seeks to determine which types of interactions contribute to students' learning. Initial analyses of CSCL processes focused on surface level characteristics of the communication, such as the number of messages sent (Strijbos et al. 2006). However, over the last 15 years elaborated analyses of communication protocols have increasingly been used to study collaboration processes (Hara et al. 2000; Rourke and Anderson 2004). These types of analyses have yielded important information about how students communicate during online collaborative learning and which kinds of communication are more conducive to learning (Strijbos et al.). By studying collaboration protocols of students chatting about historical concepts, for example, Van Drie et al. (2005) were able to demonstrate that elaborative and co-constructive communication contribute to students' learning.

Recent attempts to automatically code online collaboration

Although the study of collaboration protocols is important for furthering our understanding about how and why collaborative learning influences students' learning processes, the development of a method that can be used to analyze communication protocols can be difficult. A coding system has to be developed based on theoretical motivations and then tested (e.g., with respect to reliability and validity of the system). Furthermore, the process of analyzing a great number of protocols can be time consuming because in a typical CSCL study many groups are studied. These groups often produce extended protocols. The researcher's task becomes even more challenging when the coding scheme that is used contains more than one dimension. Van Drie et al. (2005), for example, used a coding scheme which contained four different dimensions, whereas Weinberger and Fischer (2006) even used a coding system with seven dimensions. It is not difficult to imagine that coding a large corpus of data with these kinds of elaborate coding systems is very time consuming (Dönmez et al. 2005; Rosé et al. 2008). Several researchers have therefore devoted their attention to developing techniques for automatically coding (parts or aspects of) collaboration protocols. Before discussing our own system for automatically coding collaboration protocols, we will first consider a number of other approaches.

A number of researchers have explored the possibilities of using keywords and key phrases or dictionaries to characterize the patterns of electronic communication. Such an approach assumes that in written language, users express all of their intended meaning in the written message since nonverbal cues (e.g., body language, gestures) are unavailable during such communication. Therefore, keywords or key phrases may be formulated that express certain linguistic (e.g., use of first, second, or third person pronouns) or psychological functions (e.g., expressing positive or negative emotions) of language. Computerized methods of text analysis may be used to identify occurrences of such keywords or phrases. An example of such an approach is the Linguistic Inquiry and Word Count (LIWC) system developed by Pennebaker and colleagues (cf., Pennebaker et al. 2007). Although there is evidence that the LIWC system can be used to validly measure the emotional content of written messages (cf., Alpers et al. 2005), this approach has been criticized for several reasons. First, the LIWC system has been characterized as shallow because one can question whether simply counting occurrences of words such as “grief,” “happy,” and “afraid” can be used to determine whether the author expresses positive or negative emotions without taking the context of the message into account (cf., Alpers et al. 2005; Rosé et al. 2008). A second criticism has to do with the fact that the LIWC system does not classify messages into categories; instead it provides frequencies or proportions that tell the researcher how many times the analyzed text has matched keywords or key phrases included in a LIWC category (Rosé et al.). It is, therefore, possible that the same message may match keywords in more than one category (e.g., “I hate these so-called happy occasions”). A final criticism with respect to such an approach is that the collaborative processes CSCL researchers often are interested in (e.g., depth and quality of argumentation, integration and co-construction of knowledge by multiple group members) may be too complex to be captured by isolated keywords (Rosé et al.).

An approach comparable to the LIWC system has been developed by Law et al. (2007). They have developed a CSCL discourse assessment tool called Visual Intelligent Content Analyzer (VINCA) which was used to identify keywords that were indicative of students’ cognitive and metacognitive engagement. Law et al. (2007), for example, established that keywords such as “think,” “feel,” and “believe” could be used to signal reflection during discussion. While VINCA is similar to the LIWC system in that it can give frequencies and proportions of matched keywords, Law et al. also see their system as an aid which can help researchers code their protocols. VINCA can, for example, highlight keywords or key phrases to assist the researcher during the coding process or can even assign preliminary codes to messages that can later be checked by the researcher.

The described approaches of Pennebaker et al. (2007) and Law et al. (2007) represent attempts to code protocols based on keywords and key phrases that have been defined *a priori*. Recently, other researchers have attempted to develop systems based on already *coded discourse corpora*. Goodman et al. (2005) and Soller (2004), for example, have applied computational linguistics and machine learning techniques (e.g., Hidden Markov Models, Multidimensional Scaling) to coded online interactions to train their systems to recognize effective and ineffective collaboration. A similar approach was also followed by Dönmez et al. (2005) and Rosé et al. (2008). In their research these authors attempted to develop a system that would be able to code collaboration protocols according to the system of Weinberger and Fischer (2006). They, too, used computational linguistics techniques to detect regularities in a corpus of coded discourse. Their analysis of the functioning of the system shows that such a system can yield good results when compared to messages coded by a human coder (i.e., relatively little errors were made). The work by Goodman et al., Rosé et al., and Soller shows that the CSCL community can take advantage of the advances that have been made in computational linguistics to assist in the automatic analysis of online collaboration.

Analysis of online communication based on dialogue acts

In the field of computational linguistics, a lot of work has been done in automatically recognizing, learning, or applying useful clue phrases or other linguistic characteristics (like prosody) for dialogue act tagging of spoken or transcribed utterances (e.g., Heeman et al. 1998; Hutchinson 2004; Reichman 1985; Samuel et al. 1999; Stolcke et al. 2000). The procedure described in this article aims to automatically code dialogue acts in collaboration protocols.

Our procedure for automatically coding dialogue acts started as a tool to support manual coding of dialogue acts of utterances. It has been used, elaborated, and refined in several research projects (e.g., Erkens et al. 2005; Janssen et al. 2007a). In these studies, students collaborated in small groups in a CSCL environment on research or inquiry tasks for the subjects of history and language arts. During these studies, students worked in small groups, studied information sources, constructed argumentative diagrams, and co-authored essays about their findings. Erkens et al. used the system to investigate the effects of planning tools on the process of collaboration and coordination in a CSCL environment. Janssen, Erkens, and Kanselaar used the developed system for automatic coding to give immediate feedback to students about their collaborative process. In all studies the reliability of the automatic coding was checked in comparison to manual coding. Although these studies showed the usefulness and reliability of the developed system, first for supporting manual coding and later for almost fully automatically coding dialogue acts, the validity of the system has not been systematically analyzed. As several researchers have rightly pointed out (cf., De Wever et al. 2006; Krippendorff 1980; Rourke and Anderson 2004), careful examination of the reliability and validity of any system for coding collaboration protocols is a necessary step to ensure reliable and valid research results. After describing the automatic coding procedure, the other sections of this article will, therefore, address the reliability and validity of the procedure.

The developed coding system identifies "dialogue acts," that is, the communicative function of each utterance typed by students during online collaboration and communication. The *Dialogue Act Coding* system (*DAC*) has a long history of about 20 years. The system is based on an earlier system, the Verbal Observation System, for manual coding of communicative function and content of utterances in dialogues between cooperating students. The Verbal Observation System was meant to analyze students working together on cooperative problem solving tasks (Kanselaar and Erkens 1996). The coding of communicative function and dialogue acts of the Verbal Observation System is based on research on discourse analysis by Burton (1981), and Barnes and Todd (1977), the analysis of discourse markers by Schiffrin (1987), and the analysis of question answering by Lehnert (1978). In the *DAC* system it is assumed that language users, in most cases, signal the intended meaning and interpretation of their utterances to their discourse partners by means of explicit "discourse markers" or "clue phrases" (Reichman 1985). Discourse markers are characteristic words signaling the communicative function of a phrase in conversation in natural language (Schiffrin 1987). For example, the word "*because*" at the beginning of an utterance usually indicates that the utterance is meant to be interpreted as an argumentative reason. Discourse markers are used to obtain coherence in spoken text by signaling the function of the following part of an utterance by an idiomatic phrase or word (Byron and Heeman 1997). In this way, discourse markers set up expectations of the pragmatic role that the following part of the utterance will play in the dialogue. Heeman et al. (1998) found that in a corpus of (English spoken) task-oriented dialogues, 68% of the utterances were prefixed by a discourse marker (e.g., acknowledgements, such as "oh" and "yes"). Other discourse markers can occur within the utterance itself and function internally, for example, as speech repair (e.g., "well" in Heeman et al. 1998) or as approximation (e.g., "like" in Zufferey and Popescu-Belis 2004). In the *DAC* system most discourse markers used are located at the beginning of the utterance, and mark a structural boundary or sequential dependency (Louwerse and Mitchell 2003). This way they can also be used to determine the beginning of utterances. The underlying assumption of all research on dialogue act tagging by means of discourse markers or clue phrases is that a limited set of discourse markers exists in a language, that, although dynamically changing over time and (sub) culture, are being used by speakers to establish coherence in their talk. If it is a limited set, they can in principle also be recognized by a computer system.

In the *DAC* system five main communicative functions of dialogue acts are distinguished: (1) *Argumentative* (indicating a line of argumentation or reasoning), (2) *Responsive* (e.g., confirmations, denials, and answers), (3) *Elicitative* (questions or proposals requiring a response), (4) *Informative* (transfer of information), and (5) *Imperative* (commands). We will describe the different communicative functions of dialogue acts using the example given in Table 1. This Table contains a fragment of two girls and a boy collaborating on a historical inquiry task.

Argumentative dialogue acts represent a temporal, causal, or inferential relation between utterances and use conjuncts such as “but,” “because,” and “therefore” as a discourse marker (Fraser 1999; Heeman et al. 1998). In Table 1, examples of this can be found in lines 4 and 8. In line 4, student 206 expresses a consequential line of argumentation using the word “then,” while in line 8, she gives a counterargument using the word “but.” *Responsive* dialogue acts have a backward-looking relation to an earlier utterance while the other four functions are forward looking and give new information (Louwerse and Mitchell 2003). Responsive utterances react or refer to preceding utterances (e.g., the response by student 206 in line 7). *Elicitative* dialogue acts request a response from the dialogue partner and consist of proposals to act or questions for information (Graesser and Person 1994; Lehnert 1978). In Table 1, student 204 asks what the other student is doing in line 6. The system codes this as an open question (EliQstOpn). *Informative* dialogue acts are statements transmitting new information (e.g., lines 5 and 9) or evaluations (e.g., line 11). *Imperative* dialogue acts request an action to be fulfilled by the dialogue partner. An imperative action (*ImpAct*), for instance, indicates a commanding utterance with regard to a specific action to be taken by other group members. In line 3, for example, student 204 instructs the other student to study the historical sources carefully for useful quotes. A total of 29 different dialogue acts are specified. For an overview of all the dialogue acts defined by the system, the reader is referred to Table 2.

Research questions that can be answered using the DAC system

Before describing how our system tries to automatically classify messages into the dialogue acts described above, it is useful to consider which research questions can be examined using the *DAC* system. Kanselaar and Erkens (1995, 1996) describe how the hand-coded

Table 1 Example of a coded online collaboration protocol (translated from Dutch)

Line	Student	Message	Dialogue act	Function
1	206	I’ll see if I find some quotes in the sources	InfStmAct	Inform
2	204	Yeah, ~	ResCfm	Respond
3	204	but look carefully for these quotes!	ImpAct	Command
4	206	Then we’ll name the other category, the Martyrs category	ArgThn	Argue
5	204	In every source an important person is mentioned	InfStm	Inform
6	204	And what are you doing now?	EliQstOpn	Elicit
7	206	Reading	ResRplStm	Reply
8	206	But I think we also have to make a different category about the Greek and Roman antiquity	ArgCnt	Argue
9	205	I also have sources about that	InfStm	Inform
10	204	Ok, ~	ResCfm	Respond
11	204	Good idea	InfEvlPos	Inform

t2.1 **Table 2** Overview of dialogue acts, coding categories, and examples of discourse markers (translated from Dutch)

	Communicative function	Dialogue act	Specification	Code	Description	Discourse marker, i.e.
t2.3	Argumentatives	Reason		ArgRsn	Reason, ground	“Because ...”
t2.4		Contra		ArgCnt	Counterargument	“However, ...”
t2.5	Reasoning	Conditional		ArgCon	Condition	“If ...”
t2.6		Then		ArgThn	Consequence	“Then ...”
t2.7		Disjunctive		ArgDis	Disjunctive	“Or ...”
t2.8		Conclusion		ArgCcl	Conclusion	“So, ...”
t2.9		Elaboration		ArgEla	Continuation	“Furthermore, ...”
t2.10	Responsives	Confirmation		ResCfm	Confirmation of info	“Right”
t2.11		Deny		ResDen	Refutation of info	“No”
t2.12	Reaction, or response to an utterance	Acceptation		ResAcc	Acceptance of info	“Oh”
t2.13		Reply to an elicitative	Confirm	ResRplCfm	Affirmative reply	“Sure”
t2.14			Deny	ResRplDen	Negative reply	“No way”
t2.15			Accept	ResRplAcc	Accepting reply	“Okay”
t2.16			Statement	ResRplStm	Statement reply	“ ...”
t2.17			Performative	ResRplPer	Performative reply	“Thanks”
t2.18	Informatives	Performative		InfPer	Action performed by saying it	“Hello”
t2.19	Transfer of information	Evaluation	Neutral	InfEvlNeu	Neutral evaluation	“...easy ...”
t2.20			Positive	InfEvlPos	Positive evaluation	“Nice!”
t2.21			Negative	InfEvlNeg	Negative evaluation	“Awful ...”
t2.22		Statement		InfStm	Task information	“ ...”
t2.23			Action	InfStmAct	Announcement of actions	“I’ll do ...”
t2.24			Social	InfStmSoc	Social statement	“Love you ...”
t2.25			Nonsense	InfStmNon	Nonsense statement	“grrumppphh”
t2.26	Elicitatives	Question	Verify	EliQstVer	Yes/no question	“Agree?”
t2.27	Utterances requiring a response		Set	EliQstSet	Set question/multiple choice	“.... or....?”
t2.28			Open	EliQstOpn	Open question	“Why?”
t2.29		Proposal	Action	EliPrpAct	Proposal for action	“Let’s change ...”
t2.30	Imperatives	Action		ImpAct	Order for action	“W8!”
t2.31	Commanding utterances	Focus		ImpFoc	Order group member to focus	“Hey!”

DAC system was used to gain insight into the processes that occur between students during collaborative problem solving. They studied dyads working on the "Camp puzzle," a kind of logical problem that requires students to combine different information. Kanselaar and Erkens found that dyads were mostly busy exchanging information, confirming or acknowledging their partner's contributions, and giving arguments. Contrary to their expectations, dyad members asked relatively few questions, and if they did, they were mostly aimed at checking the exchanged information. From this, Kanselaar and Erkens (1995) concluded that dyad members use different dialogue acts to coordinate their collaboration. Asking verification questions and confirming or acknowledging the partner's contribution, for example, are used as checking procedures, while argumentative dialogue acts are mostly used for negotiation of knowledge and meaning. Kanselaar and Erkens (1996) extended these findings by conducting statistical sequential analyses of the dialogues that give a deeper insight into patterns of collaboration.

Van Boxtel et al. (2000) also used an adapted version of the *DAC* system to study the effects of two different collaborative tasks (a concept-mapping task versus a poster task) on students' interaction. They found that this interaction was mostly characterized by statements, arguments, and questions. Interestingly, Van Boxtel et al. found that students who scored high on a pretest formulated more arguments during the interaction.

The system can also be used to investigate the effects of different support tools on students' collaboration and coordination. Erkens et al. (2005), for example, studied the effects of two planning tools for writing (an argumentative diagram and an outline tool). Erkens et al. found few effects of the tools on use of dialogue acts and on coordinative activities. They were, however, able to demonstrate that some coordination strategies correlated with the quality of the group texts. Use of argumentative dialogue acts during online discussion, for example, correlated positively with text quality. In conclusion, the *DAC* system may be used to answer a number of important research questions.

Automatic coding of dialogue acts

To automatically code a protocol and identify which dialogue acts are used during collaboration, the *Multiple Episode Protocol Analysis* (MEPA) computer program is used (Erkens 2005). This program can be used for the analysis and manual coding of discussions. Additionally, the program offers facilities for automatic support of coding. Over the years, a production rule system has been developed that automatically categorizes utterances into dialogue acts. A set of *if-then* rules uses pattern matching to look for typical words or phrases, in this case for discourse markers or clue phrases. Examples of discourse markers are given in Table 2. The developed production rule system consists of a rule system for automatic segmentation of combined utterances in single messages (300 rules) and a rule system for dialogue act coding (1,250 rules). In this way, MEPA is able to code a protocol consisting of 1,000 utterances in less than a second.

The rule system for segmentation of utterances (*Segmentation* filter) scans chat contributions for punctuation characters (i.e., "?," "!," "."), connectives ("however," "so," "but"), and starting discourse markers ("well," "on the other hand"). The utterance is split before or after the marker. Exception rules prevent segmentation when the same markers are used in situations that do not signal new contributions. For example, the use of full stops in abbreviations, or the non-connective uses of "but" in utterances such as "we proceed slowly *but* surely."

The *Dialogue Act Coding* filter *(DAC* filter*)* is used after segmentation of the utterances and labels messages with dialogue act codes based on recognition of discourse marking words, phrases, idiom, or partial phrases. In the *DAC* filter discourse markers are used that signify the communicative function of the message. Exception rules are used to prevent triggering of same markers that signify other functions. In the coding of responsive utterances not only discourse markers are used in the coding rules but also *information about the context of the surrounding dialogue*. A replying response is defined as a response that is referring to a preceding explicit elicitative utterance like a question or a proposal from a discourse partner other than the speaker. The production rules, therefore, check if the responsive message is preceded by a question or proposal from somebody else. In Table 1 only the responsive in line 7 is preceded by a partner's question and coded as a reply (*ResRplStm*).

If the system does not find a discourse marker in a message, the message is coded by the label InfStm? as a default catch-all. The InfStm? statements should not be confused with information statements (InfStm). The latter ones can be considered "real" information statements, whereas the former ones are statements for which the system does not find a matching discourse marker and should be checked if the message actually is an information statement. After the automatic coding, InfStm? coded messages are checked and coded manually, thus preventing erroneous coding of messages for which no known discourse marker was found. Although this means the researcher still has to check and code some parts of the protocol (about 10%), his/her job is simplified because large parts of the protocol are already coded and the uncoded parts can be easily found. However, this does not guarantee the *DAC* filter does not make any "mistakes." We will go into more detail about this in the section about the reliability of the system. Also, sometimes a "new" discourse marker is found that can be used to classify a dialogue act. This discourse marker will usually be added to the *DAC* filter.

Although automatic coding can dramatically speed up the coding process, several practical and methodological issues need to be addressed. One such practical issue concerns the language of the *DAC* filter. The filter was designed for the Dutch language, and although a procedure based on recognizing discourse markers and cue phrase can be used for other languages, it will take some time to translate the system or to develop a new one. The fact that the automatic coding procedure can only be used for content and observational variables that can be indicated by specific marker words, phrases, or actions, is another limitation of our approach. The system we propose is not suited for more interpretive variables. Researchers interested in, for example, detailed accounts of the different roles that group members perform, will find the automatic coding procedure less suitable to their needs, since these types of analyses usually involve more interpretation of students' messages and actions. A final limitation lies in the fact that the system codes most utterances in isolation; that is, the system does not take the preceding or following utterances into account when coding an utterance (the coding of replies is an exception to this). This may lead to errors because analyzing interaction often requires an interpretation of the context within which an utterance was used. When necessary, however, filter rules can be specified to take preceding or following utterances into account.

An important methodological issue, therefore, concerns the reliability of the system (Dönmez et al. 2005; Rosé et al. 2008). Of course, stability in coding is not at stake. The *DAC* filter will apply the same rules in the same manner every time and will result in the same coding for the same messages (except if new rules are added in the meantime). One aim of this article is to examine the reliability of the automatic coding system by conducting an error analysis (i.e., comparing automatic coded protocols to manually coded protocols). Another methodological issue is the validity of the coding procedure (De Wever et al. 2006; Rourke and Anderson 2004). Rourke and Anderson have outlined three types of analyses which can provide information about the validity of the automatic coding procedure. The second aim of this article is, therefore, to examine the validity of the automatic coding procedure by performing these three types of analyses.

Research questions

1. Investigation of reliability: Is the automatic coding of dialogue acts reliable when compared to manual coding of dialogue acts?
2. Investigation of validity by examining group differences: Is the automatic coding procedure able to identify differences between two different groups of language users (male and female students)?
3. Investigation of validity by examining the effects of an experimental intervention: Is the automatic coding procedure able to identify effects of an experimental intervention?
4. Investigation of validity by performing correlation analyses: Is the automatic coding procedure able to detect expected correlations with a different, manual coding system?

Method and instrumentation

Design

Data from two studies were used. For a more detailed description of both studies, the reader is referred to Janssen et al. (2007a) for Study 1, and to Janssen et al. (2007a) for Study 2. During these two studies, students collaborated in small groups in a CSCL environment on inquiry tasks for the subject of history (see the Task and Materials section below). As a part of these studies, the collaborative process between the participating students was captured in log files. While these log files were manually coded using a different coding scheme, the collaborative process can also be analyzed with the automatic coding procedure. The data collected in the two studies thus constituted a corpus of student collaboration that could be used for the analysis of the reliability and validity of the coding procedure. These studies were not set up to specifically address these issues. Rather, we saw the data from these studies, after they had been conducted, as an opportunity to address the reliability and validity of the automatic coding procedure.

Participants

Participants were 11th-grade students from several secondary schools in The Netherlands (Study 1: N=69; Study 2: N=117). These students were enrolled in the second stage of the pre-university track. Both studies were carried out in the subject of history. During the experiments students collaborated in groups of two, three, or four; students were randomly assigned to their groups by the researchers.

Tasks and materials

Students collaborated in a CSCL environment named *Virtual Collaborative Research Institute* (VCRI). The VCRI program is a groupware program designed to facilitate collaborative learning. For example, students can read the description of the group task and search for relevant information using the *Sources* tool. This information can be communicated and shared with group members, using the synchronous *Chat* tool. To write research reports and argumentative texts or essays, students can use the *Cowriter* which can be used by students to work simultaneously on the same text. Students collaborated on a historical inquiry group task for eight lessons. The groups had to use different historical and (more) contemporary sources to answer questions and co-author argumentative texts. Students were instructed to use the VCRI program to communicate with group members.

Research question 1: Examining reliability by comparing automatic and manual coding of dialogue acts

Inter-rater reliability is a critical concern with respect to the analysis of collaboration protocols and addresses the question of the objectivity of the coding procedure (De Wever et al. 2006). The important question to answer is whether if a human coder codes a protocol, he or she would assign the same codes as the *DAC* filter would (Rourke et al. 2001). To answer this question, two random segments of 500 chat messages each from Study 2 were coded using the *DAC* filter. The same 1,000 messages were coded by a human coder. This human coder was familiar with the *DAC* coding system and used the descriptions of the 29 dialogue acts to classify each chat message. The human coder interprets not only discourse markers but also the content and discourse context of the utterance to determine whether a message actually can be coded with a dialogue act code.

Research question 2: Examining validity by examining group differences

Examination of group differences can contribute to the validation of the automatic coding procedure (Rourke and Anderson 2004). The coding procedure should be able to distinguish between groups who communicate differently. For example, it has been shown quite extensively in face-to-face and computer-mediated collaboration, that women communicate differently than men do (Leaper and Smith 2004; Ridgeway 2001; Van der Meij 2007). Women are more likely to use *affiliative language* (e.g., indicating agreement, giving praise), while men are more likely to use *assertive language* (e.g., instructing others, giving arguments, indicating disagreement). Thus, the automatic coding procedure should be able to demonstrate that women use different dialogue acts than men do.

In order to demonstrate whether the automatic coding procedure was able to detect the expected gender differences during online communication, the communication of the female students in Study 1 was compared to the communication of the male students. It was expected that male students would use more assertive language during online collaboration, while female students were expected to use more affiliative language. Dialogue acts that signal affiliative language are confirmations, acceptations, and positive evaluations. Dialogue acts that signal assertive language are argumentatives, denials, negative evaluations, informative statements, and imperatives.

Of the participating students, 40 were female and 25 were male (a group of two students was excluded from the analysis because they attended only three of the eight lessons). These students collaborated in 21 different groups of which 16 were female dominated (i.e., the group consisted of more female than male students), and five were male dominated (i.e., more male than female students). The students produced 19,889 chat messages, which were automatically coded using the *DAC* filter.

Research question 3: Examining validity by examining the effects of experimental intervention

Experimental intervention may also be used to examine the validity of the automatic coding procedure (Rourke and Anderson 2004). Using this strategy, an attempt is made to modify students' behavior. It is then examined whether these changes in behavior can be detected using the instrument to be validated. During Study 1 some students had access to the *Participation tool*, whereas others did not. This tool visualizes how much each group member contributes to his or her group's *online communication* (see Fig. 1). In the Participation tool, each student is represented by a sphere. The tool allows students to compare their participation rates to those of their group members. It was assumed that the tool would influence group members' participation, and the quality of their online communication (cf., Janssen et al. 2007b) because it gives feedback about group members' participation during the collaboration (e.g., Is there equal participation in our group?) and allows them to compare themselves to other group members. This may raise group members' awareness about their collaboration and may stimulate group discussions about the collaborative process. Group members may, for example, discuss effectively planning their collaboration. Furthermore, this could draw students' attention to the quality of their discussion, and may, for example, encourage them to engage in more elaborate sequences of argumentation. On the other hand, it could be argued that by focusing students on the quantity of their contributions, instead of the quality, students could be stimulated to type more, but lower quality contributions (e.g., students try to manipulate the visualization by typing a lot of nonsense messages). This is possible, but because teachers as well as group members monitor the chat discussions, such behavior will probably be addressed.

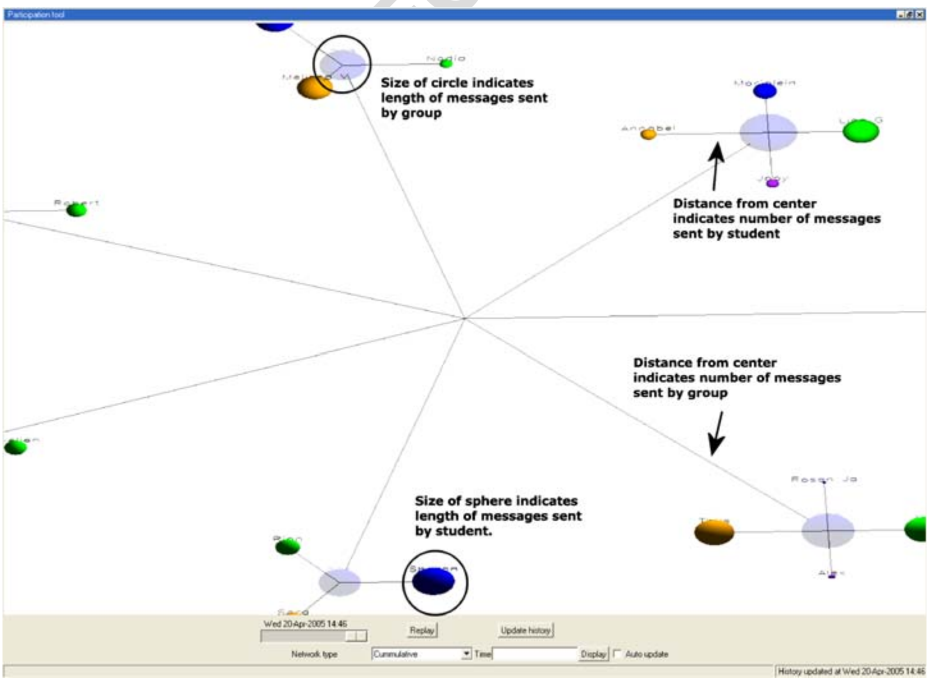

Fig. 1 Screenshot of the participation tool

In order to demonstrate the validity of the automatic coding procedure, it should be able to demonstrate differences between students with and without access to the Participation tool. When we analyzed the effects of the tool by manually coding students' collaborative activities we found that the tool stimulated students to discuss more about regulation and coordination of their collaboration, and to send more greetings. On the other hand, the Participation tool decreased students' tendency to engage in social talk (e.g., joking, swearing), decreased the number of misunderstandings between students, and decreased the number of nonsense statements (see Janssen et al. 2007b). The tool thus had an influence on the way students collaborate. The aim of this research question is to investigate whether these changes can also be detected by automatically coding the same protocols but with a focus on students' use of dialogue acts.

Research question 4: Examining validity by performing correlation analyses

Finally, correlation analyses can be used to establish the validity of the automatic coding procedure (Rourke and Anderson 2004). During such an analysis, it is attempted to demonstrate that the results of the automatic coding procedure are consistent with measurements of a similar or related construct through other methods. In order to do so, the results of the automatic coding procedure of dialogue acts were correlated with the results of a manual coding procedure of collaborative activities.

In contrast to the automatic coding procedure, the manual coding was not focused on dialogue acts (the communicative function of utterances in the dialogue of the students), but on collaborative activities (the aim and function of the utterances in the collaboration process). Whereas the dialogue act coding is based on the pragmatic, linguistic features of the utterances in the dialogue between the students (form), the coding of collaborative activities is based on the content of the utterances. The aim of this manual coding was to provide insight into the task- and group-related processes taking place between students while working together. When students collaborate on an inquiry task, they need to exchange their ideas and opinions or ask questions (Dennis and Valacich 1999; McGrath 1991). In the chat fragment provided in Table 1, lines 5 and 7 constitute examples of students exchanging information.

On the other hand, collaboration also involves a social dimension. Students have to perform social and communicative activities that establish group well-being (Kreijns et al. 2003). Therefore, the manual coding scheme also contains several codes that refer to the social aspect of collaboration, such as greeting each other, engaging in activities that contribute a positive group climate (e.g., joking, social talk), or expressing and maintaining shared understanding. In the fragment provided earlier, lines 2 and 10 constitute examples of students expressing shared understanding ("Yeah," "Ok").

Collaboration requires considerable coordination and regulation of these activities (Erkens et al. 2005). Metacognitive activities that regulate task performance (e.g., making plans, monitoring task progress) are considered important for online collaboration. Moreover, the social dimension of collaboration has to be coordinated and regulated as well (Manlove et al. 2006). For instance, students have to discuss and plan their collaboration, monitor their collaboration, and evaluate their collaborative process. Thus, the manual coding scheme for collaborative activities also contained codes that referred to the regulation and coordination of task-related and social activities. In line 6 of the fragment, student 204 asks what the other student is doing. This can be considered an example of monitoring the task process.

In total, the manual scheme contains four dimensions: *task-related activities, regulation of task-related activities*, *social activities*, and *regulation of social activities*. Each dimension contains two or more collaborative activities. In total, the scheme consists of 15 categories (see Table 3). It is important to note that the *unit of analysis* for the manual coding scheme for collaborative activities is the same as in the system for automatic coding of dialogue acts. The application of the segmentation filter precedes the actual coding (whether automatically or by hand). This ensures a comparability of the units of analysis. The manual coding of collaborative activities was done before the automatic coding of dialogue acts. The reliability of the manual coding scheme for collaborative activities has been determined in both Study 1 and Study 2 by two independent coders (Study 1: Cohen's kappa reached .86, based on 601 segments; Study 2: Cohen's kappa reached .94, based on 796 segments).

Several collaborative activities can be expected to be realized by using certain types of dialogue acts. Positive correlations between those collaborative activities and those types of dialogue acts are, therefore, expected to occur. For the current analysis, the manually coded collaborative activities were correlated with the automatically coded dialogue acts. The expected correlations between collaborative activities and dialogue acts are discussed below and are also shown in Table 3 for each collaborative activity.

First, since giving different types of arguments is considered important for effective exchange of information, argumentative dialogue acts were expected to correlate with exchange of task-related information (TaskExch). Second, because the automatically coded InfStm codes and the manually coded TaskExch codes both involve the transfer of information, a positive correlation was expected between these two codes. Third, because the manually coded TaskQues codes and the automatically coded EliQstSet and EliQstOpn pertain to questions asked by students during the collaboration process, positive correlations were expected between these codes. Fourth, because the codes MTaskEvl+, MTaskEvl−, MSociEvl+, and MSociEvl− involve either positive or negative evaluations, positive correlations were expected with the automatically coded evaluations (InfEvlPos and

Table 3 Overview of the manual coding scheme for collaborative activities and expected dialogue acts

Dimension	Collaborative activities	Expected dialogue act(s)
Performing task-related activities	Info exchange (TaskExch)	Arguments (Arg)
		Information Statements (InfStm)
	Asking questions (TaskQues)	Questions (EliQst)
Coordinating / regulating task-related activities	Planning (MTaskPlan)	Proposals (EliPrPAct)
	Monitoring (MTaskMoni)	
	Positive evaluations (MTaskEvl+)	Pos.evaluative statements (InfEvlPos)
	Negative evaluations (MTaskEvl-)	Neg.evaluative statements (InfEvlNeg)
Social activities	Greetings (SociGree)	Performatives (InfPer)
	Social support (*SociSupp*)	Pos.evaluative statements *(InfEvlPos*)
		Social Statements *(InfStmSoc*)
	Social resistance (*SociResi*)	Neg. evaluative statements *(InfEvlNeg*)
	Mutual understanding (*SociUnd+*)	Confirmations (*Res(Rpl)Cfm*)
		Acceptations (*Res(Rpl)Acc*)
	Loss of mutual understanding (*SociUnd-*)	Denials (*Res(Rpl)Den*)
Coordinating / regulating social activities	Planning (MSociPlan)	Proposals (EliPrPAct)
	Monitoring (*MSociMoni*)	
	Positive evaluations (*MSociEvl+*)	Pos.evaluative statements *(InfEvlPos*)
	Negative evaluations (*MSociEvl-*)	Neg.evaluative statements *(InfEvlNeg*)

InfEvlNeg). Fifth, during online collaboration students have to make plans and make proposals to execute a certain action. Because these types of behaviors are reflected in the manually coded MTaskPlan and MSociPlan on the one hand, and the automatically coded EliPrpAct on the other hand, positive correlations were expected. Sixth, because the automatically coded InfPer codes often involved greetings, a positive correlation was expected with the manually coded SociGree codes. Seventh, the manually coded SociSupp messages are aimed at establishing a positive group climate and often reflect positive evaluations (InfEvlPos) or social statements (InfStmSoc). Similarly, SociResi messages are detrimental to the group climate, and are thus expected to correlate with negative evaluations (InfEvlNeg). Finally, the manually coded SociUnd+ reflects the reaching and maintaining of shared understanding. This is often done by signaling acceptation or giving confirmations. Therefore, positive correlations were expected between SociUnd+ on the one hand, and ResCfm, ResRplCfm, ResAcc, and ResRplAcc on the other hand. Comparably, SociUnd- messages reflect loss of shared understanding, which is often reflected by denials. Positive correlations were, therefore, expected between SociUnd- and ResDen and ResRplDen.

As for the strength of the expected positive correlations, it should be noted that collaborative activities and dialogue acts do not refer to exactly the same construct. The collaborative activities that are distinguished can be realized by other dialogue acts as well. Weak to moderate correlations are thus to be expected.

Results

Research question 1: Examining reliability by comparing automatic and manual coding of dialogue acts

By comparing the automatically coded dialogue acts to the manually coded ones, it becomes clear that the default catch-all function of the *DAC* filter works rather well. Of the 210 disagreements (21%) between the automatic and manual coding, 106 messages (11%) were coded as InfStm? by the system. As was described earlier, the filter assigns this code to messages for which no matching discourse marker is found, leaving these messages to be checked and coded by the researcher. Although these messages are considered disagreements between the automatic and manual coding procedure in the reliability analysis presented in Table 4, their impact on research results will be limited because these messages will be checked and corrected by the researcher. The remaining disagreements are more severe because they will remain unnoticed by the researcher if he/she does not check the protocol.

Table 4 Results of the reliability analyses comparing automatic and manual coding Q1

Parameter	Reliability range *with* 'InfStm?' codes		Reliability range *without* 'InfStm?'	
Category	Agreement %	Category Kappa	Agreement %	Category Kappa
Argumentatives	0.84–1.00	0.83–1.00	0.86–1.00	0.85–1.00
Responsives	0.77–1.00	0.77–1.00	0.83–1.00	0.82–1.00
Informatives	0.04–1.00	0.04–1.00	0.12–1.00	0.12–1.00
Elicitatives	0.68–0.86	0.67–0.86	0.68–0.90	0.67–0.89
Imperatives	0.62–0.71	0.61–0.70	0.84–0.92	0.84–0.92

Table 4 reports two different calculations of the interrater reliability. The first calculation included the codes that were labeled as InfStm? by the *DAC* filter. Because the human coder did not use this code (i.e., the human coder always assigned one of the 29 codes to a message), all instances where the *DAC* filter assigned the InfStm? code can be considered as disagreements between the filter and the human coder. The first two columns reflect this calculation. However, because the InfStm? code was added to the filter to direct the researcher to cases where the filter could not assign a code based on discourse markers, one could also argue that these instances do not constitute a disagreement, because the researcher would trace these instances and subsequently code them by hand. The last two columns of Table 4 reflect this viewpoint: the instances where the *DAC* filter assigned the InfStm? code were left out of the reliability calculation.

The overall agreement between the automatic coding and the manual coding was found to be 79.0% (85.6% if the InfStm? codes are left out of the calculation), while Cohen's kappa reached 0.75 (0.84 without InfStm?). This indicates that in general the automatic and manual coding yield comparable results. This finding can be seen as evidence for the reliability of the automatic coding procedure.

Some categories of the coding system yield better results than others in terms of agreement between the *DAC* filter and the manual coding: the category Kappa's range from an unacceptable 0.04 (0.12 without InfStm?) to an excellent 1.00. When judging these kappa's against Krippendorff's standard of a minimum agreement of 0.70, 5 of the 29 coding categories fall short of this criterion (two categories without InfStm?). Remarkably, three of these five coding categories belong to the main category of *Informatives*. Especially the negative evaluations (InfEvlNeg) and nonsense statements (InfStmNon) are considerably below the 0.70 criterion. Examination of the cases where the automatic coding diverged from the manual coding yielded insight into typical "mistakes" made by the *DAC* filter. For example, in several cases the filter did not recognize words or markers that students use to signal positive or negative evaluations, and to make social statements (e.g., "this answer is chill," "this assignment is flex," "luvU"). In these cases, students often use abbreviations or expressions that they also use while chatting on MSN or sending text messages on their cell phones. Although the *DAC* filter currently does not recognize all of these phrases and words correctly, it can be updated to correctly identify these messages as well.

Table 4 also highlights the difficulties that the *DAC* filter has with recognizing nonsense statements. The 1,000-message-long protocol that was coded both automatically and by hand contained a series of over 20 nonsense messages (e.g., "blablaa," "sdvsd") typed by one student, probably because he/she was bored. In all cases, the *DAC* filter incorrectly classified these messages as information statements (InfStm). Currently, the *DAC* filter has difficulties identifying which information statements are actually meaningful and which are not. By including rules that test whether a message contains no verbs or vowels, some of the nonsense statements typed may be more easily identified by the *DAC* filter.

Another "mistake" that was sometimes made concerns confusing proposals of actions (EliPrpAct) with announcements of actions (InfStmAct), or vice versa. Sometimes, it requires interpretation or taking previous messages into account, to determine whether a message such as "I won't do it," is a proposal of actions or an announcement of actions. This confusion may also be due to an overlap between the two categories. The *DAC* filter and the human coder, furthermore, frequently disagree in the case of imperative statements, especially in the case of statements that are meant to command the focus of group members (ImpFoc). Inspection of these disagreements shows that this often happens when students use their group members' first names to gain their attention. By including a list of first names in the *DAC* filter, it might, for example, be able to better recognize these statements.

Research question 2: Examining validity by examining group differences

To determine whether male and female students used different types of dialogue acts during online collaboration, multilevel analysis was used. As students worked in groups, group level effects like gender group composition could be expected to influence the verbal behavior of individual group members. This creates a problem of nonindependence: within groups, students' scores on the variables are likely to be dependent on the scores of their group members. Traditional statistical techniques such as analysis of variance assume independence of students' dependent variables (Cress 2008). Furthermore, the nesting of students within groups creates a need to consider these two levels in the analyses (De Wever et al. 2007). Multilevel analysis can be used to deal with these issues.

A two-level multilevel model (students within groups) was constructed. In Table 5, positive betas associated with gender indicate that female students use the corresponding

Table 5 Multilevel analyses of the effect of gender on students' use of dialogue acts

Parameter	B	SE β	χ^2
Argumentatives	5.09^{**}	1.98	6.29^{**}
ArgRsn	1.20^{**}	0.48	5.85^{*}
ArgCnt	1.25	0.78	2.44
ArgCon	0.57	0.35	2.57
ArgThn	0.20	0.41	0.23
ArgDis	−0.03	0.18	0.02
ArgCcl	1.09^{*}	0.55	3.77
ArgEla	0.67	0.62	1.15
Responsives	5.91^{*}	2.70	4.60^{*}
ResCfm	5.67^{**}	2.36	5.37^{*}
ResDen	−0.09	0.29	0.09
ResAcc	0.26	0.32	0.62
ResRplCfm	0.61	0.51	1.41
ResRplDen	−0.08	0.19	0.20
ResRplAcc	0.02	0.12	0.03
ResRplStm	−0.41	0.51	0.65
ResRplPer	0.06	0.09	0.36
Informatives	-6.83^{*}	3.17	4.45^{*}
InfPer	−0.51	0.75	0.45
InfEvlNeu	0.09	0.06	2.08
InfEvlPos	0.47	0.42	1.20
InfEvlNeg	-0.51^{*}	0.26	3.79
InfStm	-7.16^{*}	3.04	5.27^{*}
InfStmAct	−0.22	0.44	0.24
InfStmSoc	0.55	0.53	1.08
InfStmNon	-0.17^{**}	0.05	9.05^{**}
Elicitatives	1.25	1.57	0.63
EliQstVer	1.48	1.11	1.74
EliQstSet	0.12	0.15	0.55
EliQstOpn	−0.64	0.65	0.96
EliPrpAct	0.37	0.46	0.65
Imperatives	-1.72^{**}	0.70	5.50^{*}
ImpAct	−0.43	0.44	0.95
ImpFoc	-1.21^{*}	0.60	3.89^{*}

$^{*}p<0.05$
$^{**}p<0.01$.

dialogue acts relatively more often than male students do. As can be seen, several effects of gender were found. Firstly, female students used relatively more argumentatives than male students did. More specifically, female students gave more reasons (ArgRsn). Furthermore, female students formulated more conclusions (ArgCcl), although the associated χ^2 only approached significance. Secondly, female students used more responsive dialogue acts. As can be seen, this is due to female students typing more confirmations (ResCfm) during online conversation. Thirdly, male students were found to use more informative dialogue acts. More specifically, male students used more informative statements (InfStm) and more nonsense informative statements (InfStmNon). Additionally, the coefficient for gender for negative evaluations (InfEvlNeg) was significantly negative. However, the corresponding χ^2-value was only marginally significant. Finally, male students were found to use more imperative dialogue acts than female students, mainly due to male students using more imperative statements which focus group members' attention (ImpFoc).

Most of the differences between male and female students were in line with our expectations. Female students used more affiliative language by typing more confirmations. Male students used more assertive language by typing more negative evaluations, informative statements, and imperatives. The result that female students used more argumentative dialogue acts was contrary to our expectation, however. Further research should investigate if this finding represents a validity problem (argumentative dialogue acts are not really argumentative) or a theoretical failure (the expectation that arguments can be seen as assertive [male] behavior is not correct). Overall, these findings show that the automatic coding procedure of dialogue acts is able to distinguish between male and female students based on their verbal communicative behavior. As such, these findings offer some support for validity of the automatic coding procedure of dialogue acts.

Research question 3: Examining validity by examining the effects of experimental intervention

It was expected that students with access to the Participation tool would participate more actively during online discussions, and that this increased participation would result in more argumentative interactions. Thus, an effect of the Participation tool was mostly expected on students' use of argumentative dialogue acts. Furthermore, because analysis with the manual coding scheme for collaborative activities (see Method section), showed effects of the Participation tool on students' use of greetings, social talk, expressions of misunderstanding, and nonsense talk, we expected these results to be mirrored in our analysis of the collaborative process using dialogue acts.

As can be seen from Table 6, these expectations were only partially confirmed. This Table does indeed show some differences between students' use of dialogue acts during online collaboration. Positive betas with respect to the condition-variable indicate that students with access to the Participation tool used the corresponding dialogue act more, compared to students without access to the tool. Concerning argumentative dialogue acts, a positive effect of the Participation tool was found on conditional arguments (ArgCon). This result should be interpreted cautiously, however, because the associated χ^2 was only marginally significant. Other differences between students with and without access to the tool were found as well.

First, students with access to the Participation tool used more confirmations in reply to elicitatives typed by group members (ResRplCfm). This mirrors the previously found effect of the tool on students' expressions of misunderstandings. Second, students with access to the tool unexpectedly replied less with statements to elicitatives (ResRplStm) typed by

Table 6 Multilevel analyses of the effect of the Participation Tool (PT) on students' use of dialogue acts

Parameter	β	SE β	X^2
Argumentatives	2.478	2.696	0.82
ArgRsn	0.121	0.602	0.04
ArgCnt	0.768	1.161	0.43
ArgCon	0.792*	0.412	3.58
ArgThn	0.092	0.529	0.03
ArgDis	−0.026	0.091	1.89
ArgCcl	0.262	0.890	0.09
ArgEla	0.126	0.739	0.03
Responsives	1.212	3.415	0.03
ResCfm	1.672	3.214	0.27
ResDen	−0.095	0.430	0.05
ResAcc	−0.390	0.381	1.04
ResRplCfm	1.246*	0.593	4.14*
ResRplDen	−0.188	0.224	0.70
ResRplAcc	−0.151	0.144	0.30
ResRplStm	−1.376*	0.585	5.31*
ResRplPer	0.028	0.109	0.06
Informatives	−8.517*	4.400	3.54
InfPer	2.239*	1.141	3.51
InfEvlNeu	−0.020	0.078	0.07
InfEvlPos	−0.447	0.561	0.63
InfEvlNeg	−0.272	0.382	0.50
InfStm	−7.907*	3.793	4.08*
InfStmAct	−0.113	0.691	0.03
InfStmSoc	−1.886*	0.838	4.54*
InfStmNon	0.112	0.075	2.10
Elicitatives	1.541	1.862	0.68
EliQstVer	0.400	1.335	0.09
EliQstSet	0.153	0.209	0.53
EliQstOpn	−0.057	0.778	0.01
EliPrpAct	1.099*	0.655	2.67
Imperatives	−1.720	0.701	5.92*
ImpAct	0.866*	0.511	2.80
ImpFoc	1.222*	0.721	2.81

* $p<0.05$

group members. Third, access to the tool had an unexpected negative impact on use of informative statements. Fourth, students with access to the tool used more performatives (InfPer), although the associated χ^2 was only marginally significant. This result mirrors the previously found effect of the tool to encourage students' use of greetings during the collaboration. Fifth, students with access to the tool used less social statements (InfStmSoc), which corresponds to our expectations. Finally, access to the Participation tool had an unexpected negative impact on students' use of imperatives.

In short, it appears that the Participation tool influenced students' online behavior, and that these changes can be detected using the automatic coding procedure. Although this detection points to the validity of the coding procedure, it should be noted that not all expected effects of the tool on students' use of dialogue acts during collaboration were found; some unexpected results were also found. More research is needed to determine whether this represents a validity problem of the automatic coding system or unexpected effects of the tool.

Research question 4: Examining validity by performing correlation analyses

Table 7 presents the correlations between collaborative activities (coded manually) and dialogue acts (coded automatically). Several positive correlations were expected (indicated by a grey background) *and* found, although most of them were weak to moderate ($r=-0.30$–0.60). As expected, several significant correlations between exchange of task-related information (TaskExch) and argumentative dialogue acts (ArgRsn, ArgCon, ArgCcl) were found. Furthermore, a significant correlation was found between exchange of task-related information (TaskExch) and information statements (InfStm). Task-related questions (TaskQues) were positively correlated with open questions (EliQstOpn), but not with the other types of the *DAC* system. In addition, positive correlations were found between making task-related and social plans (MTaskPlan and MSociPlan) and proposals for action (EliPrpAct).

Positive correlations were also expected between positive task-related and social evaluations (MTaskEvl+ and MSociEvl+) and positive evaluative dialogue acts (InfEvlPos) as well as between negative task-related and social evaluations (MTaskEvl- and MSociEvl-) and negative evaluative dialogue acts (InfEvlNeg). However, only a weak correlation between MTaskEvl+ and InfEvlPos was found.

As expected, a strong correlation between greetings (SociGree) and performatives (InfPer) was found. Furthermore, social supportive remarks (SociSupp) correlated moderately with social information statements (InfStmSoc). Additionally, because social

Table 7 Correlations between results of automatic coding of dialogue acts and manual coding of collaborative activities

	TaskExch	TaskQues	MTaskPlan	MTaskMoni	MTaskEvl+	MTaskEvl-	SociGree	SociSupp	SociResi	SociUnd+	SociUnd-	MSociPlan	MSociMoni	MsociEvl+	MSociEvl-
Argumentatives															
ArgRsn	.37**			.39**						-.36**					
ArgCnt								-.30*	-.35**						
ArgCon	.26*		.29*					-.27*		.25*		.26*		.44**	
ArgThn		-.26*	.44**									.29*		.30*	
ArgDis		.26*													
ArgCcl	.35**			.33**						-.24*					
ArgEla			.53**	.35**		-.29*									
Responsives															
ResCfm	-.44**				.24*				-.28*	.92**					
ResDen											.44**				-.27*
ResAcc															
ResRplCfm										.41**					
ResRplDen		.24*									.41**				
ResRplAcc															
ResRplStm															
ResRplPer										.34**					
Informatives															
InfPer				-.31*			.86**								
InfEvlNeu										.36**					
InfEvlPos					.30*			.53**							
InfEvlNeg		.27*						.39**	.40**	-.26					
InfStm	.56**							.32**	.24*	-.51**					
InfStmAct															
InfStmSoc	-.26*							.62**	.29**		.30*				
InfStmNon		.26*													

Cells with a grey background indicate expected correlations between dialogue acts and collaborative activities.

* $p<0.05$

** $p<0.01$

resistance remarks (SociResi) often involve negative emotions, a positive correlation was expected with negative evaluations (InfEvlNeg). Indeed, a weak correlation was found.

Shared understanding (SociUnd+) was expected to correlate positively with confirmations and acceptances. A strong correlation between SociUnd+ and ResCfm, as well as a moderate correlation between SociUnd+ and ResRplCfm was found. Similarly, loss of shared understanding (SociUnd-) was expected to correlate with denials. Indeed, weak to moderate correlations were found.

To summarize, most of the expected correlations between the automatically coded dialogue acts and the manually collaborative activities were, indeed, found. As expected, most of these correlations were weak to moderate. The unexpected correlations found, show that collaborative activities can be realized by other dialogue acts as well. With regard to validity this implies that dialogue acts and collaborative activities do not refer to exactly the same constructs and describe different aspects (related to form and to content) of communicative behavior in collaboration protocols.

Conclusions and discussion

This paper described an automatic coding procedure, which can be used to code dialogue acts in collaboration protocols. The automatic coding procedure determines the communicative function of messages. Five main communicative functions are distinguished: *argumentative* (indicating a line of argumentation or reasoning), *responsive* (e.g., confirmations, denials, and answers), *informative* (transfer of information), *elicitative* (questions or proposals requiring a response), and *imperative* (commands). A total of 29 different dialogue acts are specified.

To investigate the reliability and validity of the automatic coding procedure of dialogue acts, automatically coded dialogue acts were compared to manually dialogue acts. Although rather high kappa's were found, the analysis also showed the limitations of the automatic procedure based on recognition of discourse markers or clue phrases in utterances. Most errors were made in dynamic changing language (MSN lingo, nonsense utterances, joking) and in content- and context-defined differences using the same discourse markers. As we have explained earlier, the *DAC* filter can be changed to deal with 'new' discourse markers, but this requires the researcher to update the filter from time to time.

Additionally, we examined the validity of the automatic coding procedure using three different types of analyses. First, we examined group differences because the coding procedure should be able to distinguish between groups who are likely to communicate differently. For example, research has often demonstrated that women communicate differently than men do: women use more affiliative language, whereas men use more assertive language. The coding procedure was able to mostly replicate these findings. For example, women were found to use more responsive dialogue acts (affiliative), whereas men used more informative and imperative dialogue acts (assertive). In contrast to our expectations, we found that women used more argumentative dialogue acts. These types of utterances are often seen as assertive interactions (Leaper and Smith 2004). It remains unclear whether this finding represents a validity problem (argumentative dialogue acts are not really argumentative) or a theoretical problem (argumentative dialogue acts are not really assertive). An explanation may lie in the type of task that was employed. This task explicitly required students to discuss their findings and to exchange arguments. It might be the case that during these types of tasks, female students are more likely to engage in argumentative interactions, than during group tasks that do not explicitly call for argumentation.

Second, to examine the validity of the automatic coding procedure through examination of experimental intervention, the results of the automatic coding procedure of students with access to the Participation tool were compared to students without access to this tool. It was expected that the tool would stimulate more argumentative interactions. This expectation was only partly confirmed, as it was found that students with access to the Participation tool used more conditional arguments. However, students with access to the tool did not use more reasons, contra-arguments, etc. Again, more research is needed to determine whether this points to a validity problem of the automatic coding procedure.

Finally, results of the automatic coding procedure of dialogue acts were correlated with results of a manual coding procedure of collaborative activities. This manual coding of collaborative activities was aimed at identifying the task-related and social aspects of online collaboration based on the interpretation of the content of the utterances. Because some aspects of the manual coding procedure focused on similar or related, but not exactly the same, aspects of online collaboration as the automatic coding procedure, moderate correlations were expected. Several significant correlations were found. For example, exchange of task-related information correlated significantly with informative statements. Furthermore, making task-related and social plans were positively correlated with proposals for action. Not all expected correlations were found, however. For instance, it was expected that argumentative dialogue acts would correlate with exchange of task-related information. However, only reasons, conditional arguments, and concluding arguments correlated significantly with exchange of task-related information. The other argumentative dialogue acts did not. Not all expected correlations were found, and some unexpected correlations were found as well. This is probably due to the fact that dialogue acts and collaborative activities sometimes refer to similar but not exactly the same constructs.

In conclusion, the results we found constitute evidence in favor of the reliability and validity of the automatic coding procedure for dialogue acts. Thus, it appears the automatic coding procedure can be a useful measurement instrument for researchers who are interested in studying students' collaboration. Several questions still remain, however. First, the automatic coding procedure for dialogue acts is, for example, based on handmade production rules in contrast to approaches on automatic coding that infer the coding rules automatically from already hand-coded protocols. Obviously, specifying coding rules by hand has disadvantages, but also advantages. Disadvantages are the effort of rule construction, greater language dependency, and needed updates if language use changes over time. An advantage, in our opinion, is that the relationship to, and the continuity with, hand coding of the construct remains clear. Actually, in constructing a coding rule the human coder tries to specify explicitly the coding rules he/she implicitly uses in manual coding. Dependencies on form characteristics, content interpretation, and context knowledge become more visible. Furthermore, the automatic coding system is seen as a tool for the human coder that may support —to a lesser or greater degree—manual coding. Mixed systems, in which automatic coding and manual coding are distributed between computer and human coder depending on level of interpretation, are possible (cf., Law et al. 2007).

Second, the unit of analysis in the automatic coding procedure for discourse acts is the single message, the (part of an) utterance that conveys a single meaning. The discourse act coding of the message specifies the pragmatic, linguistic form in which this meaning is transferred and the possible communicative function that it is supposed to fulfill. Of course, the granularity and aim of the system limits its usefulness. Interpretations of communication and collaboration on higher levels, interrelating several messages to each other, such as analyses of the topic of discourse or of the development of knowledge structures, cannot be done with this system because they require interpretation of content as well.

Third, related to the previous question is the fact that the automatic coding procedure in itself does not provide insight into the structure of online discussions (Chinn 2006; Jeong 2005). It can give an overview of how many times a dialogue act was used by each group member. However, the procedure can subsequently be a starting point for more complex analyses of sequential interaction patterns (e.g., Erkens et al. 2006; Jeong; Kanselaar and Erkens 1996). These sequential analyses can subsequently be used to capture the structure and quality of online discussion.

Finally, during our validity analyses we found several unexpected results (e.g., female students unexpectedly used more argumentative dialogue acts). On the one hand, this points to the need to conduct further research to examine the validity of the automatic coding procedure. On the other hand, this does not mean the automatic coding procedure cannot be used by researchers to address their research questions. As we have shown, the system is able to code large parts of protocols reliably. Furthermore, we see the system as a valuable tool for researchers to speed up the coding process, but in some cases the researcher will need to check and sometimes correct the results of the automatic coding. As such any automatic coding procedure will probably never be able to completely replace the researcher.

Although the developed automatic coding procedure is being updated from time to time, the results of this study clearly indicate this is a suitable technique for researchers interested in the process of online collaboration. In our own research we will, therefore, try to explore the possibility to automatically code online collaboration further. For example, the outcomes of the coding procedure (i.e., the types of dialogue acts used in an online environment), can also be used as a kind of feedback to group members, giving them information about how they conduct their online discussions (Janssen et al. 2007b). Such an application of automatic coding goes beyond merely investigating how online collaboration unfolds by trying to influence collaborators to change their online behavior.

Acknowledgments This study is partly based on work carried out in a project funded by NWO, the Netherlands Organization for Scientific Research under project number 411-02-121.

References

Alpers, G. W., Winzelberg, A. J., Classen, C., Roberts, H., Dev, P., Koopman, C., et al. (2005). Evaluation of computerized text analysis in an Internet breast cancer support group. *Computers in Human Behavior*, *21*, 361–376.

Barnes, D., & Todd, F. (1977). *Communication and learning in small groups*. London: Routledge & Kegan Paul.

Burton, D. (1981). Analysing spoken discourse. In M. Coulthard, & M. Montgomery (Eds.), *Studies in discourse analysis* (pp. 61–82). London: Routledge and Kegan Paul.

Byron, D. K., & Heeman, P. A. (1997). Discouse marker use in task-oriented spoken dialogue. Paper presented at the EuroSpeech’97, Rhodes, Greece, September.

Chinn, C. A. (2006). Assessing the quality of collaborative argumentation. In S. A. Barab, K. E. Hay, & D. T. Hickey (Eds.), *Proceedings of the 7th International Conference of the Learning Sciences (ICLS)* (Vol. 2, pp. 1063–1064). Mahwah, NJ: Lawrence Erlbaum Associates.

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

Dennis, A. R., & Valacich, J. S. (1999). Rethinking media richness: Towards a theory of media synchronicity. Paper presented at the 32nd Hawaii International Conference on Information Systems (HICSS), Kohala Coast, HI.

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.

De Wever, B., Van Keer, H., Schellens, T., & Valcke, M. (2007). Applying multilevel modelling to content analysis data: Methodological issues in the study of role assignment in asynchronous discussion groups. *Learning and Instruction*, *17*, 436–447.

Dönmez, P., Rosé, C., Stegmann, K., Weinberger, A., & Fischer, F. (2005). Supporting CSCL with automatic corpus analysis technology. In T. Koschmann, D. D. Suthers, & T.-W. Chan (Eds.), *Proceedings of the 2005 conference on Computer support for collaborative learning: The next 10 years!* (pp. 125–134). Mahwah, NJ: Lawrence Erlbaum Associates.

Erkens, G. (2005). Multiple Episode Protocol Analysis (MEPA). Version 4.10. Retrieved October 24, 2005 from http://edugate.fss.uu.nl/mepa/

Erkens, G., Janssen, J., Jaspers, J., & Kanselaar, G. (2006). Visualizing participation to facilitate argumentation. In S. A. Barab, K. E. Hay, & D. T. Hickey (Eds.), *Proceedings of the 7th International Conference of the Learning Sciences (ICLS)* (Vol. 2, pp. 1095–1096). Mahwah, NJ: Lawrence Erlbaum Associates.

Erkens, G., Jaspers, J., Prangsma, M., & Kanselaar, G. (2005). Coordination processes in computer supported collaborative writing. *Computers in Human Behavior*, *21*, 463–486.

Fraser, B. (1999). What are discourse markers? *Journal of Pragmatics*, *31*, 931–952.

Goodman, B. A., Linton, F. N., Gaimari, R. D., Hitzeman, J. M., Ross, H. J., & Zarrella, G. (2005). Using dialogue features to predict trouble during collaborative learning. *User Modeling and User-Adapted Interaction*, *15*, 85–134.

Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, *31*, 104–137.

Hara, N., Bonk, C. J., & Angeli, C. (2000). Content analysis of online discussion in an applied educational psychology course. *Instructional Science*, *28*, 115–152.

Heeman, P. A., Byron, D., & Allen, J. F. (1998). Identifying discourse markers in spoken dialog. Paper presented at the AAAI Spring Symposium on Applying Machine Learning and Discourse Processing, Stanford, March.

Hutchinson, B. (2004). Acquiring the meaning of discourse markers. Paper presented at the 42nd Annual Meeting on Association for Computational Linguistics, Barcelona, Spain, July.

Janssen, J., Erkens, G., & Kanselaar, G. (2007a). Visualization of agreement and discussion processes during computer-supported collaborative learning. *Computers in Human Behavior*, *23*, 1105–1125.

Janssen, J., Erkens, G., Kanselaar, G., & Jaspers, J. (2007b). Visualization of participation: Does it contribute to successful computer-supported collaborative learning? *Computers & Education*, *49*, 1037–1065.

Jeong, A. (2005). A guide to analyzing message-response sequences and group interaction patterns in computer-mediated communication. *Distance Education*, *26*, 367–383.

Kanselaar, G., & Erkens, G. (1995). A cooperative system for collaborative problem solving. In J. L. Schnase, & E. L. Cunnius (Eds.), *CSCL'95: The first international conference on computer support for collaborative learning* (pp. 191–195). Mahwah, NJ: Lawrence Erlbaum Associates.

Kanselaar, G., & Erkens, G. (1996). Interactivity in cooperative problem solving in computers. In S. Vosniadou, E. De Corte, R. Glaser, & H. Mandl (Eds.), *International perspectives on the design of technology-supported learning environments* (pp. 185–203). Mahwah, NJ: Lawrence Erlbaum Associates.

Kreijns, K., Kirschner, P. A., & Jochems, W. (2003). Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: A review of the research. *Computers in Human Behavior*, *19*, 335–353.

Krippendorff, K. (1980). *Content analysis: An introduction to its methodology*. Beverly Hills, CA: Sage Publications.

Law, N., Yuen, J., Huang, R., Li, Y., & Pan, N. (2007). A learnable content & participation analysis toolkit for assessing CSCL learning outcomes and processes. In C. A. Chinn, G. Erkens, & S. Puntambekar (Eds.), *Mice, minds, and society: The Computer Supported Collaborative Learning (CSCL) Conference 2007* (Vol. 8, pp. 408–417). New Brunswick, NJ: International Society of the Learning Sciences.

Leaper, C., & Smith, T. E. (2004). A meta-analytic review of gender variations in children's language use: Talkativeness, affiliative speech, and assertive speech. *Developmental Psychology*, *40*, 993–1027.

Lehnert, W. G. (1978). *The process of question answering: A computer simulation of cognition*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Louwerse, M. M., & Mitchell, H. H. (2003). Toward a taxonomy of a set of discourse markers in dialog: A theoretical and computational linguistic account. *Discourse Processes*, *35*, 199–239.

Manlove, S., Lazonder, A. W., & De Jong, T. (2006). Regulative support for collaborative scientific inquiry learning. *Journal of Computer Assisted Learning*, *22*, 87–98.

McGrath, J. E. (1991). Time, interaction, and performance (TIP). *Small Group Research*, *22*, 147–174.

Pennebaker, J. W., Booth, R. J., & Francis, M. E. (2007). *Linguistic Inquiry and Word Count (LIWC)*. Mahwah, NJ: Lawrence Erlbaum Associates.

Reichman, R. (1985). *Getting computers to talk like you and me*. Cambridge, MA: MIT Press.

Ridgeway, C. L. (2001). Small-group interaction and gender. In N. J. Smelser, & P. B. Baltes (Eds.), *International encyclopedia of the social & behavioral sciences* (Vol. 21, pp. 14185–141289). Amsterdam: Elsevier.

Rosé, C. P., Wang, Y.-C., Cui, Y., Arguello, J., Weinberger, A., Stegmann, K., et al. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International Journal of Computer Supported Collaborative Learning*, *3*(3) (in press). Q2

Rourke, L., & Anderson, T. (2004). Validity in quantitative 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.

Samuel, K., Carberry, S., & Vijay-Shanker, K. (1999). Automatically selecting useful phrases for dialogue act tagging. Paper presented at the Fourth Conference of the Pacific Association for Computational Linguistics, Waterloo, Canada, June.

Schiffrin, D. (1987). *Discourse markers*. Cambridge: Cambridge University Press.

Soller, A. (2004). Computational modeling and analysis of knowledge sharing in collaborative distance learning. *User Modeling and User-Adapted Interaction*, *14*, 351–381.

Stolcke, A., Coccaro, N., Bates, R., Taylor, P., Van Ess-Dykema, C., Ries, K., et al. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, *26*, 339–373.

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.

Van Boxtel, C., Van der Linden, J., & Kanselaar, G. (2000). Collaborative learning tasks and the elaboration of conceptual knowledge. *Learning and Instruction*, *10*, 311–330.

Van der Meij, H. (2007). What has quantitative research to say about gender-linked language differences in CMC and do elementary school children's emails fit this picture? *Sex Roles*, *57*, 341–354.

Van Drie, J., Van Boxtel, C., Jaspers, J., & Kanselaar, G. (2005). Effects of representational guidance on domain specific reasoning in CSCL. *Computers in Human Behavior*, *21*, 575–602.

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

Zufferey, S., & Popescu-Belis, A. (2004). Towards automatic identification of discourse markers in dialogs: The case of like. Paper presented at the 5th SIGdial Workshop on Discourse and Dialogue, Cambridge, MA, April