Coding for Unique Ideas and Ambiguity: A Method for Measuring the Effect of Convergence on the Artifact of an Ideation Activity

ABSTRACT

Groups can generate so many ideas during a decision making process involving brainstorming that they become an impediment to group processes. Convergence activities reduce the number of ideas generated by the group and clarify those ideas, allowing the group to move forward with a set of ideas worthy of further attention. Research about convergence and its affect on collaboration is in the early stages. To further this research, measures of convergence are developed in this study as part of an assessment of the effects of convergence on an ideation artifact produced by managers attempting to solve an actual business problem. In this paper we present a method for quantifying the reduction and clarification that has occurred through convergence using an assessment of a pre- and post-convergence artifact. This study expands our understanding of collaboration by presenting the method of characterizing the convergence artifacts.

Keywords: Collaboration Engineering, Disaggregation, Convergence, thinkLets
INTRODUCTION

Groups consisting of multiple stakeholders with diverse backgrounds, varying degrees of expertise, and with differing, possibly conflicting goals must often work together to make sense of complex problems (Weick, 1993), to make decisions, and to negotiate solutions in domains such as software engineering (Boehm, Grunbacher, & Briggs, 2001; Fruhling & de Vreede, 2006), business process reorganization (den Hengst & de Vreede, 2004; Dennis, Hayes, & Daniels, 1994) and strategic decision making (Vennix, Akkermans, & Rouwette, 1996). Collaboration can be challenging, more so when decisions must be made without a clear understanding of the causes of current conditions and of potential consequences for proposed courses of action. Collaboration experts like professional facilitators, who have specialize collaboration knowledge and skills, can substantially improve group effectiveness and efficiency, but professional facilitators can be expensive, and are not always available to a group (Briggs, de Vreede, & Nunamaker, 2003). Collaboration Engineering (CE) is an approach to designing collaborative work practices for high-value recurring tasks and deploying those work practices to practitioners to execute for themselves without ongoing intervention from professional facilitators (Briggs et al., 2003).

A key goal of collaboration engineering is to distill and codify knowledge and skills into small, easily learnable concepts that non-professionals can readily use. Toward that end, CE researchers identified have six patterns of collaboration that manifest as groups work through a problem-solving process. These patterns characterized the effects of group effort as changes-of-state. The patterns are (Briggs, de Vreede, & Massey, 2008):
• Generate: move from fewer to more concepts
• Reduce: move from more to fewer concepts deemed worthy of more attention
• Clarify: move from less to more shared understanding of concepts
• Organize: move from less to more understanding of relationships among concepts,
• Evaluate: move from less to more understanding of the instrumentality of concepts toward goal attainment
• Build: commitment: move from fewer to more stakeholders willing to commit to a proposal.

Some authors combine the reduce and clarify patterns under the more general heading, *Convergence* (Davis, de Vreede, & Briggs, 2007).

A great deal has been learned about the Generate pattern of collaboration, often called brainstorming or ideation (Diehl & Stroebe, 1987, 1991; Fjermestad & Hiltz, 1998-1999; Fjermestad & Hiltz, 2001; Graham, 1977; Kolfschoten & Santanen, 2007; Lindgren, 1967; Osborn, 1963). Likewise, there are strong researcher streams about building commitment, e.g. team-building (Marks, Zaccaro, & Matthieu, 2000), negotiation (Boehm et al., 2001), and consensus building (Dunlop, 1984; Innes & Booher, 1999; Rosenau, 1962). However, the understanding of the convergence pattern is in its beginning stages.

Convergence patterns of collaboration are useful, as they often follow ideation activities which occur frequently. Often, groups generate more ideas than a group will find useful. In fact, some ideation techniques encourage group members to contribute poor ideas in addition to good ones (Osborn, 1963). A group will find it beneficial, therefore, to have a means of focusing on a reduced set of ideas. In the knowledge economy, attention may be the group’s, as well as the organization’s, scarcest resource (Davenport & Völpel, 2001). Because they free up a group’s attention for the most important issues, convergence activities become especially significant.
In order to measure the effectiveness of technology, we have to measure the value it produces for users. To evaluate technology used for convergence activities, we must be able to measure convergence effects. We therefore perform an exploratory study of technology-supported convergence. Research in this area is critical to better understandings of collaboration because activities that fall under the convergence pattern of collaboration are amongst the most difficult for facilitators to execute (de Vreede & Briggs, 2005), and group members find convergence activities to be a painful and time-consuming (Chen, Hsu, Orwig, Hoopes, & Nunamaker, 1994; Easton, George, Nunamaker, & Pendergast, 1990).

Researchers have begun to delineate concepts to explore the process and results of convergence (Davis, Badura, & de Vreede, 2008; Davis et al. 2007). Not until recently have these concepts been used to characterize convergence (Badura, Read, Briggs, & de Vreede, 2009). The current research presents this method of coding artifacts generated by groups both before and after a group has performed a convergence activity.

Specifically, we present a method of coding for the number of ideas in an artifact in order to characterize the reduction which has occurred in the artifact, as well as a method of coding the ambiguity of these ideas in order to give an indication of the clarification which has occurred as reflected in the artifact. We developed this method in order to quantify the effect of a convergence activity.

It is important that we emphasize that assessing convergence in an artifact is not as accurate as an assessment of convergence which involves the perceptions of the group who generate the ideas in the document. Perceptions of the group participating in a convergence activity are necessary to truly assess to what extent they consider the
remaining ideas worthy of further consideration and to truly assess to what extent these ideas are clarified compared to the previous version of the ideas. However, it is useful to have the ability to characterize artifacts resulting from convergence, as a researcher may more readily have access to artifacts as opposed to perceptions of the group members who generated the artifacts.

The organization of the rest of the paper is as follows. The next section contains background information relating to the two constructs we are characterizing in artifacts: ambiguity and idea reduction. In the following section, we present the procedures used in our method of characterizing the convergence. In the next two sections, we validate the coding method and present an example using the coding method with data collected from a large financial services firm and presented in "Exploring the Effects of a Convergence Intervention on Ideation Artifacts: A Multi-Group Field Study" (Badura et al., 2009). We conclude with a discussion of future research and development of the coding scheme.

BACKGROUND

In this section, we explain why it is important to accurately measure the number of ideas and whether or not these ideas are ambiguous as part of the coding scheme we present. We first explain how these constructs for assessing convergence are in line with those presented by Davis and colleagues (Davis et al., 2007) for an artifact resulting from convergence. To assess convergence, thinkLets that have convergence outcomes are assessed. A thinkLet is a codified packet of facilitation skill that can be applied by practitioners to achieve predictable, repeatable patterns of collaboration (Robert O. Briggs et al., 2003). Five constructs assess the results of a convergence thinkLet along with five assessing the process of arriving at convergence employed by the thinkLet.
the coding scheme focuses on the results of the thinkLet, we will only discuss the applicable constructs here.

- **Speed**: The length of time taken to perform the convergence activity. This may be captured in the minutes of the meeting in which the convergence activity occurred.

- **Level of Comprehensiveness**: The extent to which all ideas which the group deems worthy of further attention are included in the set which resulted from convergence.

- **Level of Shared Understanding**: The extent to which a shared context model puts boundaries around what the group is focusing on during a meeting.

- **Rate of Reduction**: The extent of the number of ideas for the group’s consideration is reduced. This can be directly assessed as a count of the ideas within the artifact used as a starting point for convergence and the resulting artifact.

- **Rate of Refinement**: The extent to which the artifact resulting from convergence is a refined, polished deliverable.

In our study, we focus on the constructs that can be assessed within a converged artifact itself. This excludes *Speed, Level of Comprehensiveness, and Rate of Refinement.* While *Rate of Refinement* and *Level of Comprehensiveness* are measurements of the artifact itself, they cannot be assessed without consulting the group which created the artifact or the individuals who will use the artifact. These were outside the scope of our study.
**Rate of Reduction** can easily be captured as a comparison of the count of ideas in the generated artifact and the converged artifact. The importance of measuring this construct is explained in the following section. **Level of Shared Understanding** is assessed through a measurement of the ambiguity of the statements in the artifact following convergence. As ambiguity decreases, the likelihood of shared understanding of the ideas within the artifact increases. The justification and importance of assessing ambiguity is presented in more detail below.

**Reduction**

A reduction in ideas results in a reduction in the number of ideas to process by a group. Theories of cognitive load describe the amount of effort expended as such processing occurs (Barrouillet, Bernardin, Portrat, Vergauwe, & Camos, 2007; Sweller, van Merrienboer, & Paas, 1998). These theories work in the context of a brain that has a short-term memory with limited processing capabilities and a long-term memory with unlimited memory abilities (Baddeley, 1992). The short-term memory is limited in its processing ability. According to the Time-Based Resource Sharing model of short-term memory (Barrouillet et al., 2007) has a bottleneck allowing only one retrieval from long term memory to occur at a time. Thus as information cues needed to be processed by accessing the long term memory increases, the time to complete the task increases. The smaller number of ideas a group has to process, therefore, equates to less work for a group to perform.

It is important to be able to measure the disaggregated ideas generated by a group, since what a group participant may forward as a single idea may, in fact, contain several ideas. Disaggregated ideas are used as the unit of analysis when considering the
relationship between such constructs as quantity and quality of ideas generated by groups (Reinig, Briggs, & Nunamaker, 2007).

**Ambiguity**

Reducing ambiguity is key to increasing shared understanding. Ambiguity describes a situation where further information is needed to make meaning more precise (Poesio, 1995). As ambiguity increases, the ability to transfer knowledge decreases (Simonin, 1999), a concept highly related to shared understanding. We define ambiguity or ambiguous as the description of a statement which can be interpreted in more than one way.

In addition to focusing the group on a smaller set of ideas, many convergence thinkLets also help the group to develop a sense of shared understanding which they did not possess beforehand. For example, in the FastFocus thinkLet, the group must come to agreement on the wording of comments proposed by individual group members. Such shared understanding of the meaning of words can be crucial in helping the group to move forward with the decision-making process. Briggs, Reinig, and Nunamaker (Briggs et al., 2003), for example, cited a case where a group negotiating requirements for a new online bookstore reached an impasse over system rights that should be granted to "affiliates." As it turned out, there were five orthogonal meanings for the word, "affiliate", in that group. Until they reached clarity on those five concepts, they could not move forward with decisions about access rights.

**THE RESEARCH VENUE**

We now present the research venue where the artifacts characterized by this coding scheme were generated. These artifacts were collected from the headquarters of a large
financial services organization. The company is over a century old and employs more than 2000 individuals.

Participants

Senior executives and associates from both the home office and the field office were involved in the workshops. Participants represented a wide spectrum of the organization both demographically and based on the departments they represented. Participants ranged in age from 37 to 50 years of age, with an average age of 42.9 years. On average, participants had 23 years of experience in this industry (Badura et al., 2009).

Task

The company hired an outside paid facilitator to design and conduct a workshop to help participants identify barriers to the organization achieving its strategic objectives. In total, six groups completed the workshop. This study describes the coding process developed to analyze the data from those sessions. The method was used to analyze data from three of those groups. While we also present a brief analysis of the data here, an analysis of the data from these sessions was reported elsewhere (Badura et al., 2009).

This study was performed by a team of four researchers. One of the researchers acted as a facilitator and enacted a scripted facilitation technique in a technology-supported meeting. This workshop included, among other activities, a brainstorming activity and a convergence activity. The facilitator did not participate in analysis of the data to avoid conflicts of interest. The development of the coding method and analysis was conducted by the other three researchers.
The group went through two workshop activities that are pertinent to this research. In the first workshop activity, the group used an idea generation technique called *FreeBrainstorm* (Briggs & de Vreede, 2001). They generated ideas for 15 minutes in response to the question, "What are the key problems that block us from obtaining our strategic objectives?"

The brainstorming technique allowed all participants to contribute simultaneously and anonymously. The first workshop yielded an artifact containing 124 comments. Subsequent sessions yielded artifacts containing 124 and 77 comments, respectively. Some of the comments contained a single problem. Some contained multiple problems. Some contained no problems. We refer to the artifact from the *FreeBrainstorm* as the *Raw Data Set*.

The second activity performed by the participants was a *FastFocus*. This activity lasted 45 minutes for each session and yielded artifacts containing 29, 31, and 20 problem statements. The artifact from the FastFocus is referred to as the *Reduced Data Set*.

*FastFocus* (Briggs & de Vreede, 2001). During this activity, each participant held a different page from the brainstorming activity. At the beginning of the activity, the facilitator displayed an empty list on a public projection screen where all participants could read it. The participants received the following instructions in accordance with the rules of the FastFocus technique:

*…Each of you is now looking at a different page. You each hold a different part of our brainstorming conversation in your hands. In a moment, I will call on each of you in turn. I will ask you, "What is the most important problem on the page in front of you that blocks us from achieving our strategic objectives?"*
The moderator then began a round robin, calling on each participant in turn. Each participant contributed a single problem statement to the public list. In accordance with the rules of FastFocus, the facilitator screened their contributions for five things:

- **Redundancy**: Participants could only add ideas that were not already on the list.
- **Relevance**: Participants could only add problem statements to the list.
- **Clarity**: Ideas had to be expressed concisely and unambiguously.
- **Levels of abstraction**: Contributions could not be so vague as to be inactionable, nor could they be so specific that they obscured the root causes of the problems.
- **Criticism**: Participants were not allowed to contribute criticisms of the ideas added by other participants. They were told they would have an opportunity to evaluate the ideas later.

When the facilitator was satisfied that a problem statement was non-redundant, relevant, clear, and at a useful level of abstraction, he typed it onto the public list. When every participant had had a chance to contribute one idea to the public list, the facilitator asked participants to switch pages. The facilitator then said, "What is the most important problem on the page in front of you that has not already been added to the public list?" Participants added new items to the list as they discovered them. The facilitator repeated the cycle of contributions and page exchanges until no participant contributed any new problem statements for three successive exchanges of pages.

The groups performed subsequent activities. These activities are outside the scope of this study and are not reported.

**PROCESS OF DISAGGREGATION**
In order to determine the effects of a convergence activity on the output of a group’s activities, a count of the number of ideas generated by the activity was necessary. An initial attempt to count the number of ideas using an aggregate method met with unacceptable results. Independent coders identified significantly different interpretations of the comments. Inter-rater reliability for one group was as low as .28. Clearly, a different method for determining the number of ideas generated by collaboration activities was needed.

Inter-rater reliability was negatively impacted by the aggregation approach. With disaggregation there is a clear stopping point. Aggregation lacks a clear stopping point. Additionally, in the initial attempt to analyze the data, the researchers were not coding for ambiguity. Since over half of the original comments were deemed to be ambiguous, this may have also interfered with inter-rater reliability.

In the revised scheme, data was coded for each group individually. Three coders were available. It was determined that two coders would analyze each group. The third coder would be available in the event that the primary coders differed in their interpretation.

To that end, the following process was enacted. The process had two phases: a disaggregation phase, where comments were separated into their component parts, and a resolution phase, where disagreements were resolved. Comments were exported from the group support system used for the workshops. The comments were imported into a spreadsheet. Each coder analyzed the comments and inserted disaggregated comments, one per cell, into the spreadsheet below the original comment. In this manner, each
coder’s interpretations ended up in columns within the spreadsheet that could be easily compared. Coders used brackets and comments to indicate ambiguity.

Definitions
The purpose of the task was to identify problem statements that met the parameters of the workshop. It was noted, however, that comments generated by participants were frequently formatted as solutions rather than problems. Therefore, researchers developed the following definitions.

Problem: A problem is a desired state or outcome that has not yet been attained (e.g., "Our customers do not feel satisfied, although we want them to.").

Symptom: A symptom is some unacceptable condition that implies some desired state or outcome that has not yet been attained (e.g., "Customers are returning products.").

Often in order for the disaggregated statement to be useful for further use, missing words needed to be included in the disaggregated statement. To that end, the researchers developed the following.

Rules
The following section describes the rules developed for the coding scheme. Each rule has a catch phrase associated with it to assist coders in recognizing the appropriate situation while coding. Comments from participants were reduced to the smallest meaningful units possible. Appendix A contains the text of the Rules of Disaggregation used by the coders (Badura et al., 2009).
Identify verbs and nouns

The first step was to identify unique noun-verb-object combinations.

**UNIQUE NVO:** Each unique noun/verb/object combination that identifies a state or outcome that has not yet been attained was disaggregated into simple problem statements. The following example shows a comment where coders identified identical unique noun-verb-object combinations:

**Original Comment:** There are too many other challenges (people, people processes, technology) that get in the way.

**Coder 1 disaggregation:**
- There are too many other challenges that get in the way
- There are too many people other challenges that get in the way
- There are too many process challenges that get in the way
- There are too many technology challenges that get in the way

**Coder 2 disaggregation:**
- There are too many other challenges that get in the way
- there are too many people challenges that get in the way
- there are too many process challenges that get in the way
- There are too many technology challenges that get in the way

When spelling and typing errors are taken into account these comments disaggregate identically.

**ACCEPT SYMPTOMS:** Identifying a symptom is an important aspect of framing a problem, so symptoms were accepted as problem statements and disaggregated using the same rules.

**MEANINGLESS VERB:** Objects were not disaggregated when doing so rendered the verb meaningless, e.g., "We feel torn between our duties to home and work." This was
not disaggregated to, "We feel torn between our duties to home" and "we feel torn between our duties to work", because doing so rendered the concept, "feeling torn between", meaningless.

ACCEPT REDUNDANCY: If people say the same thing in multiple ways in the same comment, both wordings were disaggregated into simple problem statements. Redundancy was removed in a later activity. The comment was tagged as redundant for future review.

Break phrases

BREAK OUT FIRST CAUSES from CAUSE-AND-EFFECT: When presented with a causal chain, first causes were disaggregated into standalone problem statements. For example, in the comment, "Understaffing leads to overwork, which leads to low morale", we have "understaffing" disaggregated as a separate problem statement.

DISTRIBUTE CAUSES: First causes were distributed across their consequent problem statements to make standalone problem statements. The first cause was included in the problem statement in parentheses so that the ideas could be understood in subsequent analysis steps, (e.g., rushed work) causes low satisfaction. For example, in the comment, "Understaffing leads to overwork, which leads to low morale," the understaffing cause was paired with the overwork effect, and overwork as a cause was paired with low morale, as follows: "(Understaffing) leads to overwork", and "(Overwork leads to) low morale." Thus, this comment was broken out into three problem statements.

MULTIPLE CAUSES: All first causes were combined when distributing across consequent problems, because we don't know whether either of the causes would invoke
the effect on its own. For example, in the statement, "Understaffing and overwork cause low morale", it is not possible to know whether understaffing or overwork each cause low morale, or whether both together cause low morale. Therefore, in this case, both understaffing and overwork would be broken out as first causes (see above) but low morale would be broken out, "(Understaffing and overwork) cause low morale. Thus, this comment was disaggregated into three problem statements, but they differed from the three statements illustrated in the previous rule.

NO THREE DEEP: Distributed causes will not span more than one cause and one effect. There will be no causal chains of three or more clauses in the disaggregated problem statements.

RETAIN DEPENDENT CLAUSES: Dependent clauses were not broken out as separate problem statements unless they were part of a causal chain. (Dependent clauses explain what, how, or when.) (The dependent clause may contain a problem statement when it begins with "to.")

Determine ambiguity

BRACKET AMBIGUITY: When coders found the language of the contribution allowed for multiple grammatically sound interpretations that could lead to different disaggregation structures, they made the best interpretation and disaggregated accordingly. The responses to comments that were identified as ambiguous were entered in brackets. The ambiguity was explained by stating at least two possible grammatically sound interpretations allowed by the wording of the original comment. This example
shows the two interpretations of an original comment and the coders’ indications that the comments were ambiguous.

**Original Comment:** Many classes or meetings that are scheduled around cut offs or month ends that impact production. The entire Society needs to be aware of these dates!!!!

**Coder 1 disaggregation:**

- Many classes are scheduled around cut-offs
- Many classes are scheduled around months end
- Many meetings are scheduled around cut-offs
- Many meetings are scheduled around month ends
- [Many classes are scheduled around cut-offs impact production]
- [Many classes are scheduled around months end impact production]
- [Many meetings are scheduled around cut-offs impact production]
- [Many meetings are scheduled around month ends impact production]

**Coder 2 disaggregation:**

- Many classes are scheduled around cut offs that impact production
- Many classes are scheduled around month ends that impact production
- Many meetings are scheduled around cut offs that impact production
- Many meetings are scheduled around month ends that impact production
- [The entire society needs to be aware of cutoff dates]
- [The entire society needs to be aware of month ends]

Coding ambiguous statements produced one of four outcomes: First, both coders disaggregated the original comment into two identical responses. Second, the coders each identified grammatically valid interpretations of the comment that were different but did not identify the comment as ambiguous. Third, one coder identified the comment as ambiguous while the other did not. Fourth, both coders identified the ambiguity at the time of coding. The final possibility, one or both coders identified the comment as too ambiguous to disaggregate.
BRACKET SOLUTIONS DISGUISED AS PROBLEMS: If a problem statement was rhetorically stated as a solution, coders disaggregated it into problem statements, and they used brackets to signify the ambiguity. Brackets were explained.

IGNORE THE POSITIVE: Positive clauses or phrases were not in the set of disaggregated problem statements.

RHETORICAL QUESTIONS: Statements containing rhetorical questions that can be reframed as problem statements were marked as ambiguous and reframed as problem statements. The reframed problem statements were disaggregated according to the rules.

Ways to resolve conflicts and ambiguities

Once the coders finished disaggregating the comments, the results were compared. This was the resolution phase. Any disagreements were resolved in one of four manners.

NEGOTIATE: All three coders were present for the resolution activities. Possible interpretations were discussed. If one was clearly more plausible, that interpretation was accepted and disaggregated if necessary.

SYNTHESIZE: If a one interpretation was not clearly superior, then the two coders would synthesize a better interpretation together.

ARBITRATE. The third coder was brought in to arbitrate a solution in cases where two plausible interpretations existed and the original coders could not make a determination.

DISALLOW: If coders agreed that it was not possible to determine a valid interpretation, then they would disallow the comment as too ambiguous. They did not disaggregate disallowed comments.
VALIDATION

Three raters independently coded the data from three workshops. Each group was coded by two raters with the third rater adjudicating disagreements.

Of the 70 comments found to be ambiguous by coders in Group 1, 34 were found to be ambiguous by one coder. Of the 34, 11 were found to be ambiguous by only one coder in Group 3. Of the 23 found to be ambiguous in Group 6, 13 were found to be ambiguous by one coder.

To verify the ambiguity of the comments marked ambiguous by one coder, two alternative interpretations were generated for each comment. If upon the second inspection of the comment two valid interpretations could not be generated, the comment was marked unambiguous. The original comment and alternate interpretations were presented to two new coders. The coders were instructed to code each comment that had two valid interpretations as ambiguous.

If any of the comments were not found to be ambiguous, the analysis of the comments reflected the change. The coders agreed on 89% of the ambiguities. The percentage of agreement for the three groups was 91.1%, 93.2%, and 91.7% for Groups 1, 3 and 6, respectively. Table 1 summarizes the reliability statistics for the three groups.

Percentage of agreement is not wholly sufficient as a measure of inter-rater reliability. It is too likely that the agreement is due more to chance than to agreement amongst the coders. Cohen’s Kappa examines inter-rater reliability in light of the percentage of agreement that might be due to pure chance. Cohen’s Kappa for Groups 1, 3, and 6 was
.71, .65, and .79, respectively. Cohen’s Kappa for each group showed substantial agreement on the coding of unique ideas (Landis & Koch, 1977). The Cohen’s Kappa for the ambiguity analysis was .527 reflecting moderate agreement. The lower inter-rater reliability for this analysis is likely due to the lack of clear instructions and or the significantly smaller sample size. Ambiguity is a much more subjective concept than unique idea counts. **Table 2** shows the interpretations for Cohen’s Kappa according to Landis and Koch.

```
FIELD TESTING
```
This coding scheme was used to analyze the data obtained during the aforementioned field study. A detailed explanation of the results of that analysis is reported elsewhere (Badura et al., 2009). The results are summarized here to demonstrate the measurements that can be obtained using the coding scheme. **Table 3** contains an analysis of the generated artifact as an example of the way in which the coded data can be used to develop an understanding of how a particular convergence activity affects the artifact of that activity.

```
The effects of the FastFocus activity upon the generated artifact can be observed through a comparison of the number of disaggregated problem statements and the
```
percentage of ambiguous problem statements in the generated artifact and the artifact which resulted from the FastFocus activity. We report some of the relevant statistics pertaining to the reduced artifact here to demonstrate this effect. The generated or raw artifacts contained, on average, 277 disaggregated problem statements, whereas the reduced artifacts contained only 48, on average. On average, 61% of the disaggregated comments in the raw artifacts were ambiguous while only 5% of the disaggregated comments in the reduced artifacts were ambiguous.

On average, coders identified .61 problem statements per unambiguous contribution, and 1.90 unique problem statements per ambiguous contribution.

DISCUSSION AND CONCLUSION

We present a valid method for coding the number of problem statements and number of ambiguities present in artifacts created by groups through ideation and convergence activities. This coding scheme has been shown to have strong agreement among coders for disaggregation of problem statements. This agreement carried across two sets of independent coders. This ability to measure two key indicators of group convergence activity performance is a significant contribution to the field of collaboration science. Researchers may now evaluate and improve existing convergence activities. Future research should, therefore, build upon this coding scheme by comparing artifacts generated by different convergence activities.

We found during the coding of the ambiguities that roughly 80% of them would likely have been less ambiguous had the group been trained on framing problems as different from solutions. This would include elimination of the word, "need", in specifying a lack of something as well as mentioning nouns without adjectives to
describe what about them is problematic. This emphasizes the need to further investigate the effects of brainstorming instructions on the characteristics of results. The coding scheme presented here will be helpful in this research.

The age range of workshop participants (37-50) may not have represented the company studied as a whole. However, we believe that our method of coding ambiguity is applicable to many populations in many organizations. We did not submit ambiguities discovered by both coders to be checked for multiple interpretations as described in Section 5. In future coding endeavors, a coding of all of the discovered ambiguities for multiple interpretations would give a more accurate reflection of the ambiguity of the artifact. It is, however, likely that, since both coders discovered the ambiguity independently, the statement was indeed ambiguous.

Performing and coding the disaggregation of problem statements and ambiguities took 19-21 person hours to disaggregate—two to three hours for two researchers to perform the disaggregation independently, and roughly five hours for three researchers to code the disaggregation and resolve discrepancies in disaggregation and ambiguity coding. Further effort was needed to find and eliminate redundancies. The labor involved in each effort was highly intensive. Improving the method could be a topic of research in itself. Future research should investigate the convergence measurements Speed, Level of Comprehensiveness, and Rate of Refinement, as these were not assessed in this study.

The reliability of the coding method will become more apparent as new researchers are able to follow the coding scheme with similar agreement in number of disaggregations and ambiguities as reported here.
This coding scheme could be used for any set of data generated by groups. It is also possible that this scheme would be useful for applications other than collaboration output, since none of the coding rules are inherently tied to collaboration activities.
REFERENCES


*Proceedings of the 14th Collaboration Researchers' International Workshop on Groupware*, Omaha, Nebraska.


**Appendix A**

**RULES of DISAGGREGATION**

*Source: Badura et al., 2009*

**DEFINITIONS:**

**PROBLEM:** A problem is a desired state or outcome that has not yet been attained (e.g. Our customers do not feel satisfied, although we want them to).

**SYMPTOM:** A symptom is some unacceptable condition that implies some desired state or outcome that has not yet been attained (e.g. Customers are returning products).

**GUIDELINE:** Any time you must infer missing words to make a complete noun-verb-object problem statement, add them explicitly to the problem statement in parenthesis.

**RULES:**

1. **Identify Verbs and Nouns**

   - **UNIQUE NVO:** Each unique noun/verb/object combination that identifies a state or outcome that has not yet been attained will be disaggregated into simple problem statements.
   - **ACCEPT SYMPTOMS:** Identifying a symptom is an important aspect of framing a problem, so they are acceptable as problem statements and disaggregated using the same rules.
   - **MEANINGLESS VERB:** Objects will not be disaggregated when doing so renders the verb meaningless. E.G. "We feel torn between our duties to home and work." We cannot disaggregate the problem to "we feel torn between our duties to home" and "we feel torn between our duties to work," because doing so renders the concept, "feeling torn between" meaningless.
   - **ACCEPT REDUNDANCY:** If people say the same thing in multiple ways in the same comment, disaggregate both wordings into simple problem statements. Redundancy will be removed in a later activity. Tag it as redundancy for later review.

2. **Break Phrases**
• BREAK OUT FIRST CAUSES from CAUSE-AND-EFFECT: When presented with a causal chain, disaggregate first causes into stand-alone problem statements. For example, in the comment, “Understaffing leads to overwork which leads to low morale,” “(we have) understaffing” would be disaggregated as a separate problem statement.

• DISTRIBUTE CAUSES: Distribute first causes across their consequent problem statements to make stand-alone problem statements. Add the first cause to the problem statement in parentheses so that the ideas can be understood in subsequent analysis steps. For example, in the comment, “Understaffing leads to overwork which leads to low morale,” “(we have) understaffing” would be paired with the overwork effect, and overwork as a cause would be paired with low morale, as follows: “(Understaffing) leads to overwork, and “(Overwork leads to) low morale.” Thus, this comment would be broken out into three problem statements.

• MULTIPLE CAUSES: All first causes must be combined when distributing across consequent problems, because don't know whether either of the causes would invoke the effect on its own. For example, in the statement, “Understaffing and overwork cause low morale,” it is not possible to know whether understaffing or overwork each cause low morale, or whether both together cause low morale. Therefore, in this case, both Understaffing and Overwork would be broken out as first causes (see above) but low morale would be broken out, “(Understaffing and overwork) cause low morale. Thus, this comment would be disaggregated into three problem statements, but they would differ from the three statements illustrated in the previous rule.

• NO THREE DEEP: Distributed causes will not span more than one cause and one effect - there will be no causal chains of three or more clauses in the disaggregated problem statements.

• RETAIN DEPENDENT CLAUSES - don't break out dependent clauses as separate problem statements unless they are part of a causal chain (Dependent clauses explain what, how or when) (the dependent clause may contain a problem statement when it begins with "to")

3. Determine ambiguity

• BRACKET AMBIGUITY: If you find the language of the contribution allows multiple grammatically-sound interpretations that could lead to different disaggregation structures, depending on which interpretation you adopt, make your best interpretation, disaggregate accordingly, but put your responses in brackets. Explain the ambiguity by stating at least two possible grammatically sound interpretations allowed by the wording of the original comment.

• BRACKET SOLUTIONS DISGUISED AS PROBLEMS: If a problem statement is rhetorically stated as a solution, break it out as a problem statement and bracket it. It is not possibly reliably distinguish solutions from problems during disaggregation. They can be sorted out in a subsequent activity. Explain your brackets.

• IGNORE THE POSITIVE: Do not include positive clauses or phrases in the set of disaggregated problem statements.

• RHETORICAL QUESTIONS: If a statement contains a rhetorical question that can be re-framed as a problem statement, mark the comment as ambiguous and re-frame it as a problem statement, and disaggregate it.

4. Ways to Resolution conflicts and ambiguities.
• NEGOTIATE. Discuss the possible interpretations. If one is clearly more plausible ACCEPT that interpretation and disaggregate accordingly.
• SYNTHESIZE a better interpretation together.
• DISALLOW the comment as too ambiguous if it is not possible to determine which interpretation is more plausible, DO NOT DISAGGREGATE the comment.
• ARITRATE. If coders discover that they have an irreconcilable disagreement about the meaning of the comment, submit it to a third coder for resolution.
Tables

Table 1: Percentage of agreement and Cohen's Kappa

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of Disagreements</th>
<th>Number of Disaggregated Comments</th>
<th>Number of Comments Agreed Upon</th>
<th>Percentage of Agreement</th>
<th>Cohen's Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>124</td>
<td>113</td>
<td>91%</td>
<td>0.71</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>164</td>
<td>153</td>
<td>93%</td>
<td>0.65</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>280</td>
<td>257</td>
<td>91%</td>
<td>0.79</td>
</tr>
<tr>
<td>Kappa Statistic</td>
<td>Strength of the Agreement</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;0.00</td>
<td>Poor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00 - 0.20</td>
<td>Slight</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.21 - 0.40</td>
<td>Fair</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.41 - 0.60</td>
<td>Moderate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.61 - 0.80</td>
<td>Substantial</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.81 – 1.00</td>
<td>Almost Perfect</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description of Variable</td>
<td>Variable Name</td>
<td>ID</td>
<td>Group A</td>
<td>Group B</td>
<td>Group C</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------------</td>
<td>--------------------------------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>Number of Raw Comments Contributed</td>
<td>Raw Comments</td>
<td>RC</td>
<td>124</td>
<td>124</td>
<td>77</td>
</tr>
<tr>
<td>Number of Raw Off Topic Comments (no problem statement disaggregated from the comment)</td>
<td>Raw Off-Topic</td>
<td>RC-Off</td>
<td>1</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Number of Raw On-Topic Comments</td>
<td>Raw On-Topic</td>
<td>RC- On</td>
<td>123</td>
<td>117</td>
<td>77</td>
</tr>
<tr>
<td>Number of Raw Comments deemed unambiguous</td>
<td>Raw Unambiguous</td>
<td>RCU</td>
<td>70</td>
<td>54</td>
<td>23</td>
</tr>
<tr>
<td>Number of Raw Comments deemed to be ambiguous</td>
<td>Raw Ambiguous</td>
<td>RCA</td>
<td>56</td>
<td>63</td>
<td>54</td>
</tr>
<tr>
<td>Number of Unambiguous Comments</td>
<td>Raw Ambiguity Ratio</td>
<td>A/RC</td>
<td>0.45</td>
<td>0.50</td>
<td>0.70</td>
</tr>
<tr>
<td>Number of Raw Disaggregated Problem Statements</td>
<td>Raw Disaggregated Problems</td>
<td>RCD- Problems</td>
<td>280</td>
<td>230</td>
<td>161</td>
</tr>
<tr>
<td>Number of Unique Raw Disaggregated Problem Statements</td>
<td>Raw Disaggregated Unique</td>
<td>RCD- Unique</td>
<td>246</td>
<td>220</td>
<td>154</td>
</tr>
<tr>
<td>Number of Redundant Raw Disaggregated Problem Statements</td>
<td>Raw Disaggregated Redundant</td>
<td>RCD- Redundant</td>
<td>29</td>
<td>31</td>
<td>20</td>
</tr>
<tr>
<td>Number of Raw Problem Statements disaggregated from unambiguous Raw Comments</td>
<td>Raw Disaggregated Problems - From Unambiguous Comments</td>
<td>RCD- Unambig</td>
<td>165</td>
<td>143</td>
<td>40</td>
</tr>
<tr>
<td>Number of Raw Problem Statements disaggregated from Raw Ambiguous Comments</td>
<td>Raw Disaggregated Problems - From Ambiguous Comments</td>
<td>RCD- Ambig</td>
<td>115</td>
<td>87</td>
<td>122</td>
</tr>
<tr>
<td>Average Number of Problem Statements Disaggregated from each Raw Comment</td>
<td>RC Disaggregation Ratio</td>
<td>(RCD- Unique / RC)</td>
<td>1.98</td>
<td>1.77</td>
<td>2</td>
</tr>
<tr>
<td>Average Number of Problem Statements Disaggregated from Each Unambiguous Raw Comment</td>
<td>RC Disaggregation Ratio - unambiguous</td>
<td>(RCD- Unambig / RCU)</td>
<td>2.35</td>
<td>2.65</td>
<td>1.74</td>
</tr>
</tbody>
</table>