Lightweight Techniques for Structural Evaluation of Animated Metaphors

Jorma Sajaniemi
Department of Computer Science and Statistics, University of Joensuu

Tuija Stützle
Department of Computer Science and Statistics, University of Joensuu

April 5, 2007

Abstract
Visual metaphors in the form of still or animated pictures have been used in user interfaces with the hope of enhancing learning and use of computer applications. This paper studies animated metaphors with the intent to understand how they relate to human cognition and how their quality can be measured. We present a model of the relationships within metaphors, suggest lightweight evaluation techniques based on this model, and test these techniques in an empirical investigation. The results indicate that a lightweight analysis based on still images and made by domain-aware but metaphor-unaware judges can be used as a first step in deciding which metaphors are worthy of further study, and to direct animation efforts to overcome the most crucial problems. Furthermore, the results show that animation may increase or decrease the quality of a metaphor by considerable amounts; hence the final evaluation must be based on actual use of fully implemented metaphors. The results also confirm earlier suggestions to use rich metaphors and provides evidence that richness of the still image is important for the effectiveness of animation.

Keywords: Metaphor, evaluation, animation
1 Introduction

Metaphors have been used in user interfaces with the hope of enhancing learning and use of computer applications. For example, the desktop metaphor—where the computer display is presented as a desktop with overlapping documents on it—is widely used in graphical user interfaces. As users already know that real-world documents can be transferred from one place to another, opened, closed etc., they immediately know a set of functions to expect from a user interface utilizing this metaphor. Of course, this set may not be exactly the same set that is provided by the user interface, and the way that the common functions are implemented in the user interface may be unintuitive for users (Carroll and Mazur, 1986). Thus, the benefits provided by a metaphor are by no means evident.

Metaphor involves the presentation of a new idea in terms of a more familiar one (Carroll and Mack, 1999; Ortony, 1993). In contrast to analogy, metaphor is not an exact counterpart but differs from the idea usually both in form and in content. The similarities and differences between the two ideas stimulate thought and can facilitate active learning (Carroll and Mack, 1999). However, not all dissimilarities between an idea and its metaphor give rise to active learning, and some may even hinder it. For example, the desktop metaphor gives the impression that all documents are safely located on the desk, i.e., in the computer and makes backup taking seem unnecessary (Wozny, 1989).

Metaphors may appear in several forms. Traditionally they have been verbal, but they can be pictorial, also. An example of verbal metaphors is “jail” in the phrase “My job is a jail” and an example of a pictorial metaphor is the picture of a lion used as a metaphor for power. In graphical user interfaces, pictorial metaphors are represented with static or dynamic images, e.g., a static image of binoculars representing the search function or a flying file representing transfer from one place to another. Dynamic images—or animated metaphors—are more rare than static images but their number is increasing. In the following, we will use the term visual metaphor to cover both static pictorial and animated metaphors.

Several frameworks (Alty et al., 2000; Carroll et al., 1988; Madsen, 1994; Marcus, 1998) have been suggested for the design and evaluation of user interface metaphors but none of these tackles animated metaphors explicitly. Moreover, reported uses of these frameworks are rare. In this paper, we study special properties of animated metaphors and present several lightweight techniques to evaluate the quality of visual metaphors. We also report the results of an investigation where these techniques were assessed in a software visualization context.

The rest of this paper is structured as follows. Section 2 is a literature review of metaphors and their design and evaluation techniques. Section 3 presents a model of animated metaphors and suggests several techniques to evaluate them. Section
4 describes an investigation assessing the techniques, and its results are discussed in Section 5. Section 6 concludes the paper.

2 Literature Review

This section provides an overview on previous literature on metaphors, metaphor design and metaphor evaluation.

2.1 Metaphors

Metaphor is used to explain a new or abstract thing, so called target, with the help of another familiar and usually more concrete thing, source, in order to help to get a picture or model of the target (Lakoff and Johnson, 1980). Knowledge about the target increases through interacting with the target—with the help of its metaphor—and through discussions with people familiar with it, resulting in an internal mental model of the target (Norman, 1983). A mental model explains consequences of interacting with the target and provides suggestions for potential interaction operations when some change in the target is desired.

Mental models usually are constantly evolving, technically non-accurate, and incomplete, but in order to be useful they have to be functional in some way. They also contain statements about the degree of uncertainty about the knowledge of the target. For example, after using many different calculators, people tend to develop mental models that help them to interact in the same way with all calculators even if they know that it is a suboptimal way to operate a new calculator (Norman, 1983).

Critical to the power of metaphor is that the relation between the source and the target must involve some transformation, hence people have to actively construct the relationships that comprise the metaphor (Alty et al., 2000). Salient dissimilarities of the source and the target—in the context of salient similarities—stimulate thought and can facilitate active learning (Carroll and Mack, 1999). But if a user has constant difficulties in understanding a system, he will probably feel very incompetent and stupid and will try to avoid using it in the future. According to cognitive dissonance theory (Aronson, 1997), “after engaging in a behavior that threatens the self-concept, people are motivated to reduce that threat.” If the user, however, gets some feedback or hints about how to manage with the system or finds some “treasures” of his own, he will probably get more enthusiastic about the system and feel great about himself.

Carroll and Mack (1999) stress that using metaphor activates and stimulates thought which results in better learning. In their example a user, using a computer system based on the desktop metaphor, tried to create a new document file
by sweeping the cursor across the icon representing a pad of paper, i.e., she was “tearing off paper” to obtain a new one. The metaphor was, however, misleading because a new file must be initiated through menus in this system. Having realized this, she changed her mental model accordingly. Even though one might consider the user’s struggle with the system useless, it resulted in a better mental model, i.e., better learning.

Visual metaphors rely on the effect of external visual representations to the development of mental models. It is still an open question how external representations are related to mental imagery, but the mnemonic advantage of pictorial metaphors is supported by theories of cognitive psychology (Blackwell, 2001). For example, Paivio’s dual coding theory (Paivio, 1971) predicts that recall is improved by associating an image with verbal information. Effects of visual metaphors on learners’ performance may also depend on the time allowed for mental processing of the metaphoric materials. For example, Hsu (2006) found that effects of metaphors did not appear immediately after learning but only in posttests made one day after the session that introduced the metaphors.

2.2 Approaches to Metaphors

Metaphors can be approached from three different viewpoints (Carroll et al., 1988): operational approaches focus on the learning of the target, pragmatic approaches study the interplay between the source and the target in a user’s mind in a practical work situation, and structural approaches look at the detailed similarities and dissimilarities of the source and the target. All these approaches supplement each other and are essential in designing and developing metaphors.

**Operational approach:** Operational approaches focus on the effects the metaphor has on learning new things, i.e., learning the target. For example, Mayer (1975, 1976) taught novices to program in Basic. He used two different methods: teaching with a diagram model that used familiar terms (scoreboards, ticket windows, shopping lists, . . . ) to represent the memory, input and output of a computer and compared the learning results with teaching without this model. Subjects using the meaningful model performed better on interpretations of programs and on problems requiring looping, while subjects who did not use the model performed better in straightforward problems, i.e., generation of non-looping programs.

Blackwell (2001) presents several experiments that are based on the operational approach. He used pictorial metaphors both for concepts that the participants knew beforehand (e.g., a playground slide for fall in economic market) and for concepts that were new to them (e.g., rolling balls for the dataflow “select” operation). Performance was measured by time and accuracy in problem solving tasks. The results indicate that systematic verbal and visual metaphors were less useful than consis-
tent presentation of the concepts.

Hsu (2006) used hypertext to teach details of the Internet Protocol to students. The hypertext page structure was designed to correspond to the structure of the concepts to be learned, and verbal metaphors were devised for the concepts. The effects of metaphor use (vs. hypertext materials not including the verbal metaphors) were measured by how well the concepts and their relationships were learned. This was measured in several ways: multiple-choice tests, card-sorting tasks, mental model drawing tasks, and open-ended questions—all based on the operational approach. The results suggest that metaphors may facilitate learning relationships between concepts but not learning of individual concepts. Moreover, previous domain knowledge made metaphors efficient in learning detailed information, also.

As a final example of the operational approach, Sajaniemi and Kuittinen (2005) taught Pascal programming using three different methods: in the traditional way; using a theory of the roles of variables throughout the course; and using a role-based program animator in addition to using roles in teaching. Students using the program animator, which utilizes metaphors to present roles, learned the role theory better and outperformed other groups in deep program knowledge.

**Structural approach:** Structural approaches try to explain the power of metaphors with the structural similarities and dissimilarities of the source and the target. They also measure the “goodness” of the metaphor with metrics based on structural analysis. According to Gentner’s structure-mapping framework (Gentner, 1983) analogies with the metaphor and its target can be found by defining their structures and comparing them. Structures, in this case, involve objects and their properties, relations between objects and high-order relations between relations.

Alty et al. (2000) consider necessary to list the features of a metaphor and the system for which the metaphor is used. These features are then compared to select metaphors that represent the system best. They also used ethnographic studies to find out which metaphors functioned best in various situations.

When properties of the target and the source are listed and evaluated, it is inevitable that also dissimilarities are noted. Dissimilarities are, however, not problems as such. Carroll and Mack (1999) argue that metaphors are open-ended comparisons that make use of the dissimilarities as well as similarities when they stimulate and activate thinking processes. By open-ended they mean that metaphors can not be completely defined, but there will always be different interpretations. Such rich metaphors, which have many features with some not describing their targets can be adventures to users but can also cause a lot of conceptual baggage (Alty et al., 2000). The structural approach cannot give definite answers to how well a metaphor functions but relies on empirical testing.

**Pragmatic approach:** Pragmatic approaches are not interested in the learning
of the target but in the actual usefulness of the source in practice: how well a metaphor is understood by users and how it serves the overall goals users may have. Thus, a pragmatic approach does not measure the quality of a metaphor by looking at its similarity with the target or by its effect on learning the target in general but whether it helps people to do their work in the specific context where they use the metaphor. That is, pragmatic approaches evaluate whether users learn to use the source as contrasted with learning the target which is the interest of operational approaches. For example, graphical objects in graphical user interfaces help people to understand that files and folders are places where data can be saved, and the recycle bin is a place where trash can be thrown and recovered if necessary. This knowledge concerns the metaphors themselves and requires no deep understanding of the file system concepts behind the metaphors.

The designer of a metaphor tries, of course, to find a metaphor suitable for users, but he does not live in the same environment as the users, does not know all specialities of their work context, and his good intentions may have failed. Empirical investigations may show how much the dissimilarities between the metaphor and the target distract users and causes them cognitive dissonance; or if the dissimilarities actually do the opposite: activate users to explore new functions of the system in the actual use situation.

### 2.3 Design and Evaluation of Metaphors

Several frameworks for designing and evaluating user interface metaphors have been presented in literature. The oldest, suggested by Carroll et al. (1988) consists of four steps: identify candidate metaphors or composite metaphors; detail metaphor/software matches with respect to representative user scenarios; identify likely mismatches and their implications; and identify design strategies to help users manage mismatches. This basic framework has been further developed by Alty et al. (2000), Marcus (1998), and Madsen (1994).

Alty et al. (2000) divide the process into six steps: identify system functionality; generate and describe potential metaphors; analyze metaphor-system pairings; implementation issues (representation, realism, consistency); evaluation; and feedback on design. Identifying system functionality is necessary in order to be able to generate some alternatives for metaphors. Users’ environments, previous systems and the language they use can give some hints for appropriate metaphors. Brainstorming is one technique to invent many possible suggestions for metaphors. Also ethnographic techniques, which describe the knowledge and understanding that the future users of the system have of their own world, articulated in their own linguistic and conceptual terms, can be useful.

In the analysis of metaphor-system pairings, the features of the metaphor and
the system are divided into three categories of interest: S+M+ are features that are found both in the system (S+) and in the metaphor (M+) showing the amount of the system functionality mapping onto the core features of the metaphor; S+M- are features found only in the system showing the extent to which the system functionality fails to map onto the metaphor; and S-M+ are features found only in metaphor possibly prompting incorrect assumptions about the system and resulting in extra conceptual baggage. The extent of features in these categories help the designer to see problems and possibilities of the metaphor. Rich metaphors, which have many features, will always have some features not mapping onto the system. Despite this, Alty et al. suggest the use of metaphors that encourage users to explore new possibilities of the system: although it can cause extra cognitive load in the beginning, it can lead to better user performance in the future.

The next step is to decide about implementation issues: how realistic metaphors should be, are metaphors consistent with respect to others, and are they appropriate (e.g., culturally acceptable) for users. In the evaluation step, the system with its metaphors is to be tested in the real working environment with real tasks. The purpose of the evaluation is to see how suitable the metaphor is considering the amount of conceptual baggage, misunderstandings, inappropriateness of metaphors, extensibility, consistency and degree of realism. Based on the results the design is then improved. During the whole design process and especially during the evaluation, direct feedback to the re-design of the prototype interface is collected.

Marcus (1998) divides the process of defining user interface metaphors into five steps: identify items among data and functions that should be targets; identify sources of metaphorical reference; generate many possible metaphors; identify and evaluate matches and mismatches; and revise metaphors to strengthen effective matches and reduce harmful mismatches. The second and third steps together form Alty et al.’s second step, but the most important difference to Alty et al. is the stress on studying similarities and dissimilarities in the evaluation part. Marcus describes a case study, a travel agent system to be used worldwide. Several versions of the user interface were built to determine the style, color, size and spatial organization of the metaphoric images, means of highlighting, and the atmosphere (“happiness”). Feedback from discussions with users and the client’s product development managers was used to make improvements to the design.

In contrast to metaphor design, Madsen (1994) studies metaphorical design where a metaphor is selected first and the design of the system is guided by the metaphor. For example, Madsen describes a project where services provided by research libraries were described with three different metaphors: the warehouse, the store and the meeting place. These metaphors pointed out three different possibilities what libraries are, leading to three different kinds of computer applications. Most of Madsen’s guidelines concerning metaphor generation apply also when de-
signing metaphors, e.g., advice for finding alternatives—building on already existing metaphors, using artifacts of previous systems as metaphors, using already implicit metaphors, and looking for real-world events exhibiting key aspects—can be used as design principles, also. When evaluating metaphors Madsen encourages to select a metaphor with rich structure and to validate that it covers all relevant aspects of the system. The metaphor should be familiar to and well-understood by the users, but it can also have some conceptual distance to the target in order to help the users to see things in a new way.

3 Animated Metaphors

Current technology provides the possibility to use animation in visual metaphors resulting in animated metaphors. This section provides a model of animated metaphors consisting of the target, various aspects of the animated source, and their relationship to designers’ and users’ mental models. Then, this model is expanded so that it can be used as a basis for lightweight structural evaluation of visual metaphors.

3.1 Design of Animated Metaphors

Visual metaphors are usually implemented as still pictures, e.g., the scissors symbol used as a metaphor for the cut\(^1\) operation of Prolog (Eisenstadt and Brayshaw, 1988), but animation has also been used, e.g, in the dancing animals metaphor for values of variables in visualizing algorithms (Hübscher-Younger and Narayanan, 2003) or the animated mouse cursor effects of current graphical user interfaces. Animation adds a new dimension to metaphor design because the focus of animation may be the source of the metaphor, the target, or both.

For example, in the Prolog scissors metaphor an animated flash at the edge of the blade would underline that the metaphor is a pair of scissors, i.e., the focus of the animation would be on the source. On the other hand, an animated transfer of open scissors over the program execution control lines and breaking them one by one while the open scissors slide across the lines would emphasize the target, i.e., the effect of the cut operation on the control flow of the Prolog program. Finally, a movement of the scissors over the control lines with the blades opening and closing for each control line would emphasize both the source and the target simultaneously.

\(^1\)For a reader unfamiliar with computer programming concepts, there is an appendix at the end of the paper that gives a short glossary of programming related terms.
One may assume—even though this has not been verified—that the effect of animation depends on its focus. If the source is the focus, animation may not increase learning of the target but just help in recognizing the source or give enjoyment to the viewer. Focusing on the target is supposedly more efficient for understanding the target more thoroughly. Finally, focusing on both the source and the target may emphasize the connection between the source and the target and may increase a sensation of analogy possibly resulting in too strict expectations of the similarity of the source and the target.

Partial support for this claim is provided by two studies. Hübscher-Younger and Narayanan (2003) found that the presence of metaphoric content led to higher pleasure but lower salience; and that the use of an unconventional metaphor led to higher pleasure and originality. In their study, animation was used to underline the source, e.g., that animals are living creatures. On the other hand, Sajaniemi and Kuittinen (2005) used animation to emphasize the target—dataflow between variables—and they found that the use of metaphors increased students’ understanding of how to use variables in programs.

In the design framework suggested by Alty et al. (2000), animation is not treated explicitly but it falls in the implementation issues step. However, the focus aspect must be considered earlier in the analysis of metaphor-system pairings. For example, animating S+M+ features may overemphasize the similarity of the system and the metaphor. Moreover, S+M- features that have potential for animation should be analyzed carefully because an animation whose focus is on the target might be used to emphasize that the system actually has the animated feature even though the metaphor lacks it.

Figure 1 somewhere here

Figure 1 gathers up the relationships between the source and the target of an animated metaphor when the source consists of two parts, the image and the animation, and the focus of the animation may be the source or the target, or both. The target has usually some name which may be a metaphor, also. For example, in the scissors metaphor the target is the abstract Prolog operation for abandoning control branches and its name, “cut”, is also a metaphor. The selection of the name has an effect on the quality of the visual metaphor, e.g., if the name of the operation were “forget”, another metaphor than scissors might function better. The designer’s mental model is affected by the target and the designer tries to represent his or her mental model by the metaphor, i.e., both the name and the source are selected by the designer based on his or her understanding of the target. Both the name and source affect the mental model of the target developed by a user, and this mental model should represent the target as accurately as possible.
3.2 Lightweight Structural Evaluation Techniques

The quality of a metaphor can be measured by the accuracy of the user’s mental model: how well does it correspond to the target? Direct measurement of this accuracy using operational or pragmatic approaches means that users must work with the metaphor in a real setting. Such techniques are expensive because they require full implementation of the metaphor, a large enough group of users that know the domain and have the ability to understand the target, and time for the users to learn the target using the metaphor. Subjects willing to participate in such an evaluation may be hard to find, and with animated metaphors the implementation is not a trivial task. Moreover, if several metaphors are to be compared, the same users cannot be used because of the mental interference caused by the sources that were studied earlier (Anderson, 2000, pp. 207–221). Therefore, simpler and cheaper evaluation techniques are desirable.

If the full implementation and extensive use of a metaphor are to be avoided, assessment must be based on structural approach: a comparison of the similarities and dissimilarities of the source and the target. The simplest way to do this—and the one implicitly suggested in the design frameworks discussed in Section 2—is to assign the task to the designers of the metaphor. In terms of the notions described in Figure 1, this technique means that the designer compares his or her own mental model of the target with the name and source created by himself or herself. However, people are poor in finding information that would falsify their expectations (Anderson, 2000, pp. 330–332). As the designers know the similarities of the target and the source that have guided the design of the source, and as they expect the metaphor to be good, they are poor in finding dissimilarities between the target and the source. Moreover, the designers are using their own mental model of the target—not the target itself—which makes the comparison biased.

These problems can be attacked by not using the designer as the only person in metaphor evaluation. We will now introduce several lightweight techniques that use two groups of people participating in the evaluation: subjects—representing users in the evaluation—who find and list structural properties of the source, and independent judges—replacing the designer—who evaluate the similarity of the listed properties with properties of the target. This can be done in different ways that will be described in more detail below (use Figure 2 for reference).

Subjects may list properties of the source based on the full source, i.e., the image and the animation, but isolated from the target, i.e., without using the metaphor in its actual context. In fact, if the focus of the animation is on the source, subjects
need not understand the target or be aware of its existence at all: they simply look at the source image or watch it animate, and list its properties with no reference to any domain context at all. Consequently, there are no requirements on the knowledge of the target or even of any specific domain for persons used as subjects in the evaluation. Moreover, this technique allows the use of the same subjects for the evaluation of alternative metaphors for the same target. The subjects’ task is simply to report properties of various sources; they do not work with the sources nor do they relate the sources to any target, and thus no mental interference effects do occur. Only priming effects must be taken care of, i.e., the order of presentation of various sources may affect a subject’s interpretation of sources presented later, but such priming effects can be handled easily by using breaks at appropriate points. In Figure 2 properties extracted in this way are labeled as “perceived properties of source”.

If the implementation of animation is laborious, this type of evaluation can be based on the properties of the still image part of the metaphor. Excluding the animation part of the source has the additional advantage of setting even smaller constraints on subjects. Animation whose focus is even partially on the target requires understanding of the target domain whereas still images of common things can be understood basically by anyone. In Figure 2 properties extracted in this simple way are labeled as “perceived properties of image”.

Sometimes several sources are designed together to form a family of images or animations with the hope that they will reinforce the whole and contribute to the proper interpretation of individual metaphors, e.g., forks, spoons, knives etc. in a cutlery metaphor (Blackwell, 2001). In such a case, all sources may be presented to a subject at the same time, and the subject may be allowed to list properties in any order. By seeing all sources at the same time, subjects can detect discriminating properties of each source—in the same way as these properties affect users’ mental models of the targets.

The similarity of perceived properties and the properties of the target may then be assessed by independent judges, who code the perceived properties as similar, dissimilar, or neutral to the target or its properties. It is possible that the judges are target aware and thus have their own mental model of it. In Figure 2 such a model is labeled as “independent mental model of target”. Target-aware judges may also know the suggested metaphors and may unconsciously mix properties of the source with properties of the target. In order to avoid such favoring of planned metaphors, the assessment of perceived properties can be done by independent target-aware domain experts that are not acquainted with the planned source.

In cases, where such independent domain experts are unavailable, independent target-unaware judges may also be used. They can base their decisions on condensed descriptions of the target. In Figure 2 “descriptive properties” refer
to a description given by the designer, and “salient properties” refers to a very short description abstracting the most central properties of the target. It should be noted that both descriptive and salient properties are listed by the designer of the metaphor and they have probably guided the designer in designing the metaphor, also. Thus these properties may unconsciously favor the metaphor that is to be evaluated. An alternative way would be to ask an independent target-aware (but metaphor-unaware) expert to write them.

When several targets that are related within a domain are assessed at the same time, target-aware judges are able to distinguish more important properties of each target from less important ones and they may use this knowledge when assessing perceived properties of the sources. Target-unaware judges do not know all targets beforehand and they may err in interpreting, e.g., the first target description they see. To help target-unaware judges to discriminate between targets, they can be given all target descriptions before they start their assessment task.

In metaphor evaluation, subject perceived properties (based on full source or still image only) can be assessed by judges having any of the above knowledge (designer’s mental model of the target, independent mental model of the target, descriptive properties of the target, or salient properties of the target). It is clear that such a technique, e.g., comparing salient properties of the target with perceived properties of the still image of the source, discard some information and cannot provide an “optimal” evaluation. On the other hand, individual preferences differ and it is doubtful that metaphors everybody would call perfect do exist. Therefore, it may be useless to strive for a perfect evaluation technique but rather acquiesce in a simple technique that is able to detect some of the largest problems. The next section describes an investigation where several of the above subject/judge combinations were used and their results were compared with metaphor evaluation results obtained by operational and pragmatic approaches.

4 Investigation

To test the lightweight structural techniques described in the previous section, we conducted an investigation by studying animated metaphors used in a computer program animator. In an earlier study (Stützle and Sajaniemi, 2005), these metaphors have been studied using the operational and pragmatic approaches, i.e., evaluated from the users’ perspective. These user acceptance results are now used as a measure of the mental model accuracy that will be compared with judges’ evaluations of the similarity of subjects’ perceived properties of images with properties of the target. Especially, we want to see how judges’ evaluations should be interpreted and to what extent they do reflect users’ mental model accuracy.
We will start by describing the metaphors and their acceptance from the users’ viewpoint in Subsection 4.1. Then we will describe the current investigation, and present its results in Subsection 4.2. Discussion of the results is postponed to Section 5.

4.1 User Acceptance of the Metaphors

The investigation was carried out in the context of metaphors used in a computer program animator. The animator (Sajaniemi and Kuittinen, 2005) is based on the concept of roles of variables—a piece of programming knowledge that can be taught explicitly to novices. The motivation for the use of metaphors in the animator is their anticipated positive effects on learning and, in fact, the use of the program animator has been found to improve students’s programming skills considerably (Byckling and Sajaniemi, 2006).

Table 1 gives informal definitions of the roles used in the investigation; all roles with their formal and informal definitions can be found in the Roles of Variables Home Page at http://www.cs.joensuu.fi/~saja/var_roles/.

<table>
<thead>
<tr>
<th>Role</th>
<th>Informal Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home Page</td>
<td>Table 1 somewhere here</td>
</tr>
</tbody>
</table>

The program animator utilizes roles in two ways. First, each variable is depicted by a role image that visualizes the role, e.g., variables having the role fixed value are depicted by a stone giving the impression of a value that cannot be changed. Second, the animation of operations depends on the roles. For example, an assignment to a follower is animated by transferring the value of the followed variable into the follower. The animation of comparisons depends on the role, also. For example, Figure 3 gives visualizations of the two syntactically similar comparisons “some_variable > 0”. In case (a), the variable is a most-recent holder, e.g., an input value given by the user, and the comparison just checks whether the value is in the allowed range that appears on the screen when the comparison is made. In case (b), the variable is a stepper with several values shown within the role image. In this case, no new values appear; only the colors of the existing values change and the user can see the result by the color of the current value.

| Figure 3 somewhere here |

In summary, role images are supposed to reflect how variables having that role behave, i.e., role images are used as visual metaphors for the roles, and the focus of animation of operations is most often the target, i.e., the role.
In the earlier study (Stützle and Sajaniemi, 2005), the original metaphors were contrasted with control metaphors (see Figures 4 and 5), which were deliberately chosen not to directly reflect role properties so that clear distinctions could be expected. Two versions of the animator were used, one with role metaphors and the other with control metaphors, and animation was controlled by using similar animation effects in both versions.

The participants, thirteen second year computer science students, had earlier encountered roles in their studies but they had not used the animator. After a short recap of the roles, participants had to assign roles to 14 variables in a pretest. Based on its results, participants were divided into original and control groups that performed equally well in the pretest. In the study, the original group evaluated the original role metaphors and the control group the control metaphors. First the participants reported their perceived properties of the images. Then they used the appropriate version of the animator after which their recognition of the metaphors was tested and they were asked to grade the metaphors. The session was concluded with a posttest on role knowledge: again, participants were asked to assign roles to 14 new variables.

The results of assigning roles to variables at the pretest and posttest are given in Table 2. Based on Wilcoxon’s rank sum test, the difference in change is significant \( p = 0.0012 \). Thus the original metaphor set facilitated learning of the target concepts when compared with the control metaphors, i.e., taken as a whole, the original metaphors were better in the sense of the operational approach.

Pragmatic approach was used to evaluate individual metaphors in two respects: how well participants learned the metaphors, and how appropriate participants found them. Figure 6 presents the proportion of participants recognizing each metaphor after use of the animator. The original group recognized the roles distinctively better. Based on Fisher’s exact test, the differences between the groups were significant for stepper \( p = 0.0210 \) and follower \( p = 0.0006 \) and almost significant for fixed value \( p = 0.0699 \), gatherer \( p = 0.0775 \) and temporary \( p = 0.0862 \).
Table 3 presents means of grades that the participants gave to the metaphors. The grades are given in the Finnish school grading system 4–10 with 4 being the worst and 10 the best grade. These figures include only grades by participants that recognized the role correctly. Based on two-tailed t test the difference between the groups is significant for stepper ($t = 5.131$, $df = 7$, $p = 0.0014$) and fixed value ($t = 2.909$, $df = 8$, $p = 0.0196$) but not for temporary. For follower and gatherer statistical testing cannot be done because so few participants of the control group recognized these metaphors correctly.

A more detailed description of the method and results of the earlier study can be found in Stützle and Sajaniemi (2005).

4.2 Method

In the previous study, participants’ acceptance of the metaphors was measured in several ways providing data that can be used as a measure of the accuracy of users’ mental models. Before using the animator, i.e., when the participants did not yet know that the images were metaphors, they listed adjective and verb associations for the images, i.e., they acted as subjects of the lightweight metaphor evaluation techniques introduced in Subsection 3.2. These associations are now used as perceived image properties that will be evaluated by judges.

Thus, in the current investigation, judges (or coders) code the perceived image properties with respect to the targets and their properties, and we want to see how judges’ evaluations should be interpreted and to what extent they do reflect users’ mental model accuracy. In addition to this validity aspect of the applied techniques, we will also study their reliability.

The current investigation was a between-subjects design with the target knowledge type and judges’ background as the between-subjects factor. The five conditions were (see Figure 2):

- judges with knowledge of roles and the animator, i.e., having designer’s mental model of target (target-metaphor knowledge, TM)
- judges with knowledge of roles but not of the animator, i.e., having independent mental model of target (target, TA)
- domain-aware (but target-unaware) judges that were given descriptive properties of the target (descriptive/domain, DD)
• domain-unaware judges that were given descriptive properties of the target
  (descriptive/non-domain, DN)

• domain-unaware judges that were given salient properties of the target
  (salient, SA)²

Judges: There were a total of fifteen judges, three in all conditions. They will
be denoted in the following with the condition abbreviation followed by a number,
e.g., the three judges in the target-metaphor condition are denoted by TM-1, TM-2,
and TM-3; the three target condition judges by TA-1, TA-2, and TA-3, etc.

In the target-metaphor condition, the judges were doctoral students of the role
theory and they were acquainted with the animator and its metaphors. The judges
in the target condition were computer science educators that were given a 1 hour
introduction to the role theory; they did not know about the animator or its role
images. In the descriptive/domain condition the judges were fourth to sixth year
computer science students, and in the descriptive/non-domain and salient condi-
tions the judges were first to third year humanities students.

Materials: Before using the animator, participants of the previous study gave
adjective and verb associations for each of the role images (either original or con-
control images depending on the group). These words were collected in five lists, one
for each role, and sorted alphabetically. A total of 350 different words were given,
varying from 64 to 81 for each role. Thus the materials consisted of five word lists,
one for each role, containing adjective and verb associations given by all subjects
to either the original or control image.

Lists were titled depending on the condition. In the target-metaphor and target
conditions, titles were simply the names of the roles. In the descriptive/domain
and descriptive/non-domain conditions, titles were one paragraph descriptions of
the target. These descriptions made no reference to programming but were ex-
pressed in general terms. For example, Figure 7 depicts the word list for stepper in
these two conditions. In the salient condition, titles were few words that made no
reference to programming, e.g., for stepper: “predetermined, regular”.

Procedure: The judges coded all words in the word lists. Words were coded
as relating positively (marked with “+”), negatively (“−”), or not related to the title
of the word list. Judges were instructed to make another pass through a word list, if
they did mark considerably less than half of the words in the first round; however,

²The salient condition was run as a part of the previous study, but for the sake of clarity it will be
presented as a part of the current investigation.
they were not forced to mark half of the words on the second round. There was no restriction on the proportions of plus and minus marks. There was no time limit; the durations varied between 22 and 45 minutes.

In two conditions, target-metaphor and descriptive/non-domain, judges coded the materials a second time two weeks after the first coding. These codings were used for intracoder reliability analysis.

4.3 Results

The results provide information about three aspects of the word coding: intracoder reliability, intercoder reliability, and the relationship between judges’ evaluation of subjects’ word lists and the evaluation of users’ mental model accuracy in Subsection 4.1.

**Intracoder Reliability:** In two conditions, target-metaphor and descriptive/non-domain, judges coded the materials twice with two weeks interval in between. Table 4 lists the proportion of different change types for each word over all six judges. In most cases (76.1%) a judge gave the same code to a word on both times, and in only 3.0% of cases the code given at the second coding was totally opposite to the earlier given code. In the rest of the cases, it was more common to switch from a neutral attitude to non-neutral (i.e., positive or negative) than vice versa. Overall, changes between neutral and negative were more frequent (11.6%) than between neutral and positive (9.3%).

Cohen’s kappa is a commonly used statistical measure for rater agreement. It is more robust than simple percentage as it takes into account the agreement occurring by chance. A complete agreement yields the value of 1; a perfectly random agreement yields zero. Table 5 gives Cohen’s kappa values for the two codings of each of the judges that coded the same materials two times. The smallest kappa value, i.e., poorest agreement, is obtained for the judge TM-3 because this judge coded a large number of words (12.6%) as neutral in the first time and as negative at the second coding two weeks later. The major source for the next lowest kappa value (judge DN-3) is due to a large number of words (10.0%) coded first as negative and two weeks later as neutral.

In the following, only the first coding made in the target-metaphor and descriptive/non-domain conditions will be considered when we compare intercoder reliability among all judges. In other conditions, each judge coded the materials only once.
**Intercoder Reliability:** Overall, the fifteen judges coded 53.3% of the words as neutral, 27.4% as positive, and 19.3% as negative.

Table 6 somewhere here

Table 6 gives kappa values for codings of all judge pairs. Kappa values for pairs of judges in the same condition are in boldface.

To see how individual words were coded, a disagreement measure was defined. For example, if a word was coded as positive by 7 judges, neutral by 5 judges, and negative by 3 judges, disagreement is $5 \times 1 + 3 \times 2$ (i.e., 5 judges at “distance” 1 and 3 judges at “distance” 2 from the majority of judges). Even though only 6.0% of words were given the same code by all fifteen judges, a majority of the words (60.9%) had disagreement level 5 or less. Thus, at least ten judges had used the same code for a majority of the words.

**Metaphor Evaluation:** Figures 8–12 give the proportion of words judges coded as having positive (bars above x-axis) or negative (below x-axis) relation to the associated target. For example, the first two bars in Figure 8 show that the three judges in the target-metaphor condition (TM) found on average 38% of the words given to the original *stepper* image relating positively and 28% relating negatively to the *stepper* concept, and 4% of the words given to the control image relating positively and 15% negatively to *stepper*.

Figures 8, 9, 10, 11, and 12 somewhere here

Taking all roles together, the proportion of non-neutral associations was greater for the original metaphors than the control metaphors. This was true both for positively related words (paired $t$ test, $t = 5.968, df = 24, p < .0001$) and for negatively related words ($t = 3.898, df = 24, p = .0007$). The relationship between judges’ evaluation and the mental model accuracy, i.e., user acceptance, will be analyzed in detail in the next section.

**5 Discussion**

An evaluation technique must satisfy two requirements in order to be useful: it must be reliable, i.e., produce the same result every time used with the same data set, and valid, i.e., produce the correct result. We will now look at these aspects of the lightweight structural, judge-based techniques introduced in subsection 3.2, and consider also the validity of the investigation itself.
Reliability of the Lightweight Evaluation Techniques: The Cohen’s kappa values in Table 5 indicate good intracoder reliability (Neuendorf, 2002, p. 143). Low kappa values were mostly due to changes between neutral and negative coding among the two codings accomplished two weeks apart. This means that intracoder reliability was better for words coded positive than for words with negative codes.

In intercoder reliability (Table 6), there is a large variation of kappa values. If .400 is considered to be the limit between poor and fair agreement, the salient condition (SA) must be rejected. That is, a few words description of the target is not sufficient for judges. Indeed, the reliability of the salient condition is so poor, that it will not be considered in the rest of this discussion.

The best kappa values are among judges in the target condition (TA). However, they indicate fair agreement only, and it seems advisable to use several judges in order to compensate for individual preferences. The kappa values in the target-metaphor condition (TM) are somewhat lower and they have larger variation. This may be due to unconscious use of knowledge of the implemented metaphor: one judge may be more capable of not using the knowledge than another. Thus, knowing the target but not the suggested metaphor seems to produce best intercoder reliability and fidelity to the technique.

Validity of the Lightweight Evaluation Techniques: The important question is how well the accuracy of users’ mental model can be predicted based on judges’ coding of subjects’ perceived properties of the images. In the following, the user acceptance described in Subsection 4.1 and measured by recognition (Figure 6) and grade (Table 3) is taken as the measure of the accuracy of the mental model (see Figure 2) for each of the metaphors. Our interest here is to find out whether this accuracy can be predicted by utilizing judges for assessing subject perceived properties of still images.

The original metaphor for follower is a stereotype of a good metaphor: the number of positive associations is large and the number of negative associations is small (black bars in Figure 10), its recognition is excellent (black bar for stepper in Figure 6) and grade is good (stepper/original in Table 3). Thus the accuracy of users’ mental model is very good and the structural evaluation based on word coding reveals a large number of similarities and a small number of dissimilarities between the source and the target. This is just what one would expect.

A small number of negative associations is, however, not a must: the original metaphor for stepper has a large number of negative associations (black bars in Figure 8) even though it has excellent recognition and grade. With this metaphor the animation clearly demonstrated the predictability of the value sequence and probably led thus to an accurate mental model reflected in the excellent recognition and grade. The animation was, however, presented to the participants only after they had reported properties of the still images. The reported properties included a large
amount of words relating to uncertainty because of footprints going to two directions. The judges coded such words as negative associations because the behavior of a stepper is predictable and regular. The animation, however, demonstrated certainty and had thus the potential to strengthen effective matches and reduce harmful mismatches (Marcus, 1998). This probably made recognition easier and explains the contradiction between mismatching structural properties of the still image and the target, and excellent recognition and grade of the metaphor as a whole. Moreover, the image (footprints) has a clear connection with the verbal metaphor of the role, i.e., its name (stepper). The excellent recognition and grade are probably affected by this double metaphor effect, also.

The control image for stepper had the same animation as the original image, but its recognition and grade were poor. Thus a good animation alone does not make a poor metaphor acceptable even though it may increase the accuracy of the resulting mental model. A notable difference between judges’ coding of the words for the original and control metaphors for stepper (Figure 8) is that the original image has a large amount of both positive and negative associations whereas the control image has a small number of them. This suggests a possible explanation for the fact that the animation improved the accuracy of the original metaphor but did not do that for the control metaphor: the larger amount of positive associations of the original metaphor made it possible for users to find suitable positive associations that could be strengthened by the animation. In the case of the control metaphor, there simply might have been so few matches between the source and the target that none of them could be strengthened by the specific animation. This suggests that the effect of animation depends on the richness of the metaphor.

The control metaphor for temporary has similar amount of positive and negative associations (light bars in Figure 12) as the original metaphor for follower (black bars in Figure 10). Its recognition is fair and grade poor, while the original metaphor for follower has excellent recognition and good grade. The focus of the follower animation is on the target whereas the focus of the temporary metaphor is mainly on the source. Thus two metaphors with the same amount of positive and negative associations may differ in quality depending on the animation and its focus. Especially, focusing animation on the target seems to lead to a better mental model of the target.

The control metaphors for stepper (light bars in Figure 8), follower (light bars in Figure 10) and gatherer (light bars in Figure 11) have a small number of both positive and negative associations and their recognition and grades are poor. In general, the proportion of non-neutral associations was smaller for the control metaphors than for the original metaphors (positively related words \( p < .0001 \), negatively related words \( p = .0007 \)), and the recognition and grades of all control metaphors were poor. This suggests that a metaphor with a small amount of both
positive and negative associations in judges’ coding may be expected to be poor. When the number of non-neutral associations is small, the metaphor is in fact not related to the target in any way and thus does not provide matches and mismatches that would enhance learning. This finding supports suggestions (Alty et al., 2000; Carroll and Mack, 1999; Madsen, 1994) to use rich metaphors.

In summary, a simple structural evaluation of still images seems to provide useful information about the quality of metaphors. Especially, the richness of a metaphor demonstrated by a large number of non-neutral associations seems to indicate a good metaphor since it provides users many hooks where they can hang their knowledge about the target. Even though positive associations are definitely needed, a large number of negative associations may be compensated by animation effects. However, animation may also decrease metaphor quality depending on its target. Therefore, an evaluation based purely on still images is unreliable. This leads to the need of more expensive and laborious techniques based on full implementation of metaphors. However, cheap and simple structural evaluation of still images can be used as a first step in deciding which metaphors are worthy of further study. Furthermore, negative associations found in still images may give hints on improving metaphors and they can be used to direct animation efforts to cases where animation support is needed.

**Validity of the Investigation:** This study has a number of limitations: the number of metaphors is small, the focus of animation has not been varied in a systematic way, users’ learning time has been short, and perceived properties have been gathered for still images only. On the other hand, the metaphors have been evaluated in a real program animation environment with participants representing its real users and judges representing persons typically available for software visualization evaluation.

It is clear that some discrepancies between a metaphor and its target are more serious than others, and it is hard to tell which differences are serious and which are not. To attack this problem, the evaluation of judges has in this investigation been compared with the quality of users’ mental models. Thus, the assessment of the lightweight evaluation techniques is not a theoretical one but is rooted on measured user acceptance. Even though it is impossible to say which individual words (i.e., perceived properties) are particularly good or bad for the quality of the metaphors, the connection between overall patterns of judges’ word coding and metaphor quality has been clarified by this investigation. In fact, we have not looked at coding of individual words but only at coding patterns of all words for each metaphor.
6 Conclusion

In this paper, we have studied animated metaphors with the intent to understand how they relate to human cognition and how their quality can be measured. We have presented a model of the relationships between the source and the target of an animated metaphor, discussed the role of the focus of animation for the quality of animated metaphors, and developed lightweight techniques for structural evaluation of metaphors. Finally, we have conducted an empirical investigation assessing several lightweight techniques in the context of a program animator and users learning programming concepts by working with new metaphors.

Our model of metaphor relationships comprises the source and the target, the name of the target, the designer’s and user’s mental models of the target, and the relationships among these. The source consists of an image and animation that can be focused on the source, the target, or both. We have hypothesized that the effect of animation depends on its focus, and we have found partial support for this. If the source is the focus, animation may not increase learning of the target but rather help in recognizing the source or give enjoyment to the viewer. Focusing on the target is supposedly more efficient for understanding the target more thoroughly. Finally, focusing on both the source and the target may emphasize the connection between the two and result in even too strict expectations of the similarity of the source and the target.

Based on the model, we have introduced lightweight techniques for structural evaluation of metaphors. These techniques use two groups of people participating in the evaluation: subjects find and list structural properties of the source, and judges evaluate the similarity of the listed properties with properties of the target. We have considered several ways of presenting information about the source to subjects, and about the target to judges, and found that different ways place different requirements for subjects and judges, e.g., on their knowledge about the target or the domain of the target. Then, we have studied empirically how judges’ evaluations should be interpreted and to what extent they do reflect users’ mental model accuracy.

The results indicate that a structural analysis based on still images and made by domain-aware but metaphor-unaware judges can be used as a first step in deciding which metaphors are worthy of further study, and to direct animation efforts to overcome the most crucial problems. Furthermore, the results show that animation may increase or decrease the quality of metaphor by considerable amounts; hence the final evaluation must be based on actual use of fully implemented metaphors. The results also confirm earlier suggestions to use rich metaphors, i.e., metaphors with a large number of matches and mismatches between the source and the target. In particular we found cases where the same animation improved the quality of
a rich metaphor but did not do that for another, non-rich metaphor for the same target.

It must, however, be remembered that metaphor design is not a scientific endeavour but rather an art, and many evaluation techniques—including those presented in this paper—fail to capture this aspect. Much of the quality of a metaphor depends on its visual design and details of animation. Some discrepancies between the source and the target are more serious than others, and this difference is not recognized by the evaluation techniques suggested above. Thus, there is always the danger of rejecting excellent metaphors just because their excellency is not recognized by the technique used.

We have not considered metaphor design in detail but mostly relied on literature on this issue. We have, however, presented the new notion of the focus of animation and discussed its effects on metaphor quality—an issue that must be considered when designing an animated metaphor. There are also several limitations of the empirical investigation. Notably, the focus of animation was not systematically varied in the investigation and hence its effects could not be tracked thoroughly. Further research into the effects of the focus is needed.

Acknowledgments

This work was supported by the Academy of Finland under grant number 206574.

References


## Appendix: Glossary of Programming

This appendix lists computer programming related terms used in the article.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>The operation of setting a new value to a variable.</td>
</tr>
<tr>
<td>Comparison</td>
<td>The operation of determining whether two values (obtained from expressions containing variables and/or constants) are equal, or if not, which is larger.</td>
</tr>
<tr>
<td>Control flow</td>
<td>The order in which individual statements are executed.</td>
</tr>
<tr>
<td>Control line</td>
<td>Visual representation of a branch in a Prolog program derivation tree that represents the goal hierarchy of the program with certain input values.</td>
</tr>
<tr>
<td>Cut</td>
<td>Prolog predicate that eliminates subgoals, or choices, in a Prolog program derivation tree.</td>
</tr>
<tr>
<td>Dataflow</td>
<td>Flow of data from variables to other variables determined by assignments executed in the order determined by control flow.</td>
</tr>
<tr>
<td>Program</td>
<td>A list of statements meant to be executed by a computer.</td>
</tr>
<tr>
<td>Program execution</td>
<td>The activity of a computer performing the statements of a computer program in the proper order.</td>
</tr>
<tr>
<td>Role of a variable</td>
<td>The execution-time behavior of a variable described by a name, e.g., <em>stepper</em>.</td>
</tr>
<tr>
<td>Variable</td>
<td>A place in computer memory where a quantity can be stored; usually referred to by a name in a computer program.</td>
</tr>
</tbody>
</table>
Figure 1: A model of the relationships between the source and target of an animated metaphor.

Figure 2: Properties that can be used in structural evaluation of an animated metaphor.
Figure 3: Visualizations of the same operation for different roles: comparing whether a most-recent holder (a) or a \textit{stepper} (b) is positive.

Figure 4: Original and control metaphors for \textit{stepper} (Stütze and Sajaniemi, 2005).

Figure 5: Original and control metaphors for \textit{fixed value}, \textit{follower}, \textit{gatherer}, and \textit{temporary} (Stütze and Sajaniemi, 2005).
A “stepper” proceeds in a predictable way to some known direction. The progress obeys some clear rule and is most often smooth. Ordinarily the progress has some known destination.

- pace [word given by some subject(s) to the original image]
- proceed [word given by some subject(s) to the original image]
- hesitate [word given by some subject(s) to the original image]
- hesitating [word given by some subject(s) to the original image]
- uncertain [word given by some subject(s) to the original image]
- departing [word given by some subject(s) to the original image]
- chop [word given by some subject(s) to the control image]
- split [word given by some subject(s) to the control image]
- burst [word given by some subject(s) to the control image]
- funny [word given by some subject(s) to the original image]

Figure 7: An excerpt of the word list for stepper given to the judges in the descriptive/domain (DD) and descriptive/non-domain (DN) conditions. The one paragraph description of the target is followed by the perceived properties of the two stepper images. The explanations in square brackets were not included in the word lists presented to judges. Original order of words that is based on alphabetical order of the Finnish words is preserved.
Figure 8: Proportion of positive and negative associations for *stepper*.

Figure 9: Proportion of positive and negative associations for *fixed value*.

Figure 10: Proportion of positive and negative associations for *follower*.
Figure 11: Proportion of positive and negative associations for *gatherer*.

Figure 12: Proportion of positive and negative associations for *temporary*.
Table 1: Roles of variables used in the investigation.

<table>
<thead>
<tr>
<th>Role</th>
<th>Informal description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stepper</td>
<td>A variable stepping through a systematic, predictable succession of values.</td>
</tr>
<tr>
<td>Fixed value</td>
<td>A variable initialized without any calculation and not changed thereafter.</td>
</tr>
<tr>
<td>Follower</td>
<td>A variable that gets its new value always from the old value of some other variable.</td>
</tr>
<tr>
<td>Gatherer</td>
<td>A variable accumulating the effect of individual values.</td>
</tr>
<tr>
<td>Temporary</td>
<td>A variable holding some value for a very short time only.</td>
</tr>
</tbody>
</table>

Table 2: Performance of the groups in the pretest and posttest. Scale 0-14; 14 being best (Stützle and Sajaniemi, 2005).

<table>
<thead>
<tr>
<th>Group</th>
<th>Pretest</th>
<th>Posttest</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>6.7</td>
<td>11.4</td>
<td>4.7</td>
</tr>
<tr>
<td>Control</td>
<td>6.7</td>
<td>5.8</td>
<td>-0.9</td>
</tr>
</tbody>
</table>

Table 3: Means of role metaphor grades. Scale 4–10; 10 being best (Stützle and Sajaniemi, 2005).

<table>
<thead>
<tr>
<th>Group</th>
<th>Stepper</th>
<th>Fixed value</th>
<th>Follower</th>
<th>Gatherer</th>
<th>Temp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>9.3</td>
<td>8.9</td>
<td>8.0</td>
<td>8.8</td>
<td>5.8</td>
</tr>
<tr>
<td>Control</td>
<td>5.5</td>
<td>6.7</td>
<td>5.0</td>
<td>5.0</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Intracoder code changes from the first coding to the second coding two weeks later.

<table>
<thead>
<tr>
<th>Change type</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remained same</td>
<td>76.1%</td>
</tr>
<tr>
<td>From neutral to positive or negative</td>
<td>12.2%</td>
</tr>
<tr>
<td>From positive or negative to neutral</td>
<td>8.7%</td>
</tr>
<tr>
<td>From positive to negative or vice versa</td>
<td>3.0%</td>
</tr>
</tbody>
</table>
Table 5: Kappa values for intracoder reliability analysis.

<table>
<thead>
<tr>
<th>Judge</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM-1</td>
<td>.721</td>
</tr>
<tr>
<td>TM-2</td>
<td>.772</td>
</tr>
<tr>
<td>TM-3</td>
<td>.370</td>
</tr>
<tr>
<td>DN-1</td>
<td>.660</td>
</tr>
<tr>
<td>DN-2</td>
<td>.610</td>
</tr>
<tr>
<td>DN-3</td>
<td>.496</td>
</tr>
</tbody>
</table>

Table 6: Kappa values for intercoder reliability analysis.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TM-1</td>
<td>.456</td>
<td>.344</td>
<td>.337</td>
<td>.447</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TM-2</td>
<td></td>
<td></td>
<td></td>
<td>.472</td>
<td>.485</td>
<td>.512</td>
<td>.504</td>
<td>.524</td>
<td>.482</td>
<td>.483</td>
<td>.494</td>
<td>.482</td>
<td>.483</td>
<td>.482</td>
<td>.482</td>
</tr>
<tr>
<td>TA-1</td>
<td>.310</td>
<td>.490</td>
<td>.358</td>
<td>.322</td>
<td>.358</td>
<td>.351</td>
<td>.322</td>
<td>.358</td>
<td>.351</td>
<td>.322</td>
<td>.358</td>
<td>.351</td>
<td>.322</td>
<td>.358</td>
<td>.351</td>
</tr>
<tr>
<td>TA-2</td>
<td>.342</td>
<td>.438</td>
<td>.370</td>
<td>.312</td>
<td>.333</td>
<td>.359</td>
<td>.312</td>
<td>.333</td>
<td>.359</td>
<td>.312</td>
<td>.333</td>
<td>.359</td>
<td>.312</td>
<td>.333</td>
<td>.359</td>
</tr>
<tr>
<td>DD-1</td>
<td>.230</td>
<td>.389</td>
<td>.256</td>
<td>.322</td>
<td>.358</td>
<td>.351</td>
<td>.322</td>
<td>.358</td>
<td>.351</td>
<td>.322</td>
<td>.358</td>
<td>.351</td>
<td>.322</td>
<td>.358</td>
<td>.351</td>
</tr>
<tr>
<td>DD-2</td>
<td>.353</td>
<td>.418</td>
<td>.229</td>
<td>.312</td>
<td>.333</td>
<td>.359</td>
<td>.312</td>
<td>.333</td>
<td>.359</td>
<td>.312</td>
<td>.333</td>
<td>.359</td>
<td>.312</td>
<td>.333</td>
<td>.359</td>
</tr>
<tr>
<td>DN-1</td>
<td>.393</td>
<td>.421</td>
<td>.349</td>
<td>.393</td>
<td>.426</td>
<td>.434</td>
<td>.393</td>
<td>.426</td>
<td>.434</td>
<td>.393</td>
<td>.426</td>
<td>.434</td>
<td>.393</td>
<td>.426</td>
<td>.434</td>
</tr>
<tr>
<td>DN-3</td>
<td>.338</td>
<td>.397</td>
<td>.264</td>
<td>.324</td>
<td>.381</td>
<td>.338</td>
<td>.324</td>
<td>.381</td>
<td>.338</td>
<td>.324</td>
<td>.381</td>
<td>.338</td>
<td>.324</td>
<td>.381</td>
<td>.338</td>
</tr>
<tr>
<td>SA-1</td>
<td>.152</td>
<td>.210</td>
<td>.075</td>
<td>.079</td>
<td>.139</td>
<td>.138</td>
<td>.223</td>
<td>.257</td>
<td>.152</td>
<td>.079</td>
<td>.132</td>
<td>.189</td>
<td>.079</td>
<td>.132</td>
<td>.189</td>
</tr>
<tr>
<td>SA-2</td>
<td>.140</td>
<td>.213</td>
<td>.148</td>
<td>.199</td>
<td>.176</td>
<td>.162</td>
<td>.219</td>
<td>.186</td>
<td>.192</td>
<td>.217</td>
<td>.304</td>
<td>.228</td>
<td>.243</td>
<td>.211</td>
<td>.228</td>
</tr>
<tr>
<td>SA-3</td>
<td>.199</td>
<td>.201</td>
<td>.171</td>
<td>.189</td>
<td>.280</td>
<td>.161</td>
<td>.066</td>
<td>.151</td>
<td>.211</td>
<td>.229</td>
<td>.217</td>
<td>.169</td>
<td>.156</td>
<td>.211</td>
<td>.169</td>
</tr>
</tbody>
</table>