

An Empirical Evaluation of Visual Metaphors in the Animation of Roles of Variables

Tuija Stütze and Jorma Sajaniemi
University of Joensuu, Joensuu, Finland

tstutze@cs.joensuu.fi jorma.sajaniemi@cs.joensuu.fi

Abstract

Roles of variables, which describe stereotypic usages of variables, can be exploited to facilitate teaching introductory programming. This paper describes the evaluation of visual metaphors for roles used in a role-based program animator. The evaluation is based on several criteria: properties of the images, metaphor recognition and grading, and effects on learning. The study demonstrates that as a whole the role metaphors facilitate learning. The results also identify ideas for further elaboration of the individual metaphors. Furthermore, the study suggests that the evaluation of animated metaphors may require special measures.

Keywords: Roles of variables, metaphor, program animation, computer science education

Introduction

Learning to program is hard to many students. To facilitate teaching introductory programming Sajaniemi (2002) has developed a theory about the roles of variables—a piece of programming knowledge that can be taught explicitly to novices. Only ten roles are needed to cover 99 % of all variables in novice-level programming, and they can be described in a compact and easily understandable way. Moreover, program animation can be based on roles, which provides a possibility to elaborate students' role knowledge in a meaningful way.

Kuittinen and Sajaniemi (2004) have described how roles can be introduced and exploited in teaching programming. They conducted a classroom experiment where students were taught programming in three different ways: in the traditional way in which the course had been given several times before, i.e., with no specific treatment of roles; using roles throughout the course; and using roles together with the use of a role-based animator in exercises. The introduction of roles was found to provide students a new conceptual framework, which enabled them to mentally process programs in a way similar to that of good code comprehenders. Moreover, the use of role-based animation in exercises seemed to assist in the adoption of role knowledge and expert-like programming skill.

The role concept has a strong cognitive foundation. Sajaniemi & Navarro Prieto (submitted) in-

Material published as part of this journal, either online or in print, is copyrighted by the publisher of Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Editor@inform.nu to request redistribution permission.

vestigated experts' programming knowledge in a knowledge elicitation study. They found several types of variable-related knowledge in expert programmers, including roles. There was some variation in role boundaries and in the granularity of roles, but roles could anyhow be easily detected in participants' knowledge. Thus, roles represent expert programmers' tacit

knowledge even though individual differences in details do exist. Ben-Ari and Sajaniemi (2004) demonstrated that in one hour's work, computer science educators can learn roles as defined by Sajaniemi (2002), and assign them successfully in normal cases. Thus, the selected set of roles is natural and can be easily adopted by CS teachers.

Sajaniemi and Kuittinen (2003) have presented a program animator, PlanAni, which utilizes roles in two ways. First, each variable is depicted by a role image that visualizes the salient properties of the role. For example, 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. Role images are used to reflect the way variables having that role behave, i.e., role images are used as visual metaphors for the roles.

Metaphor involves the presentation of a new idea (target) in terms of a more familiar one (source) (Carroll & Mack, 1999; Ortony, 1993). Metaphors differ from analogies in that an analogy is functionally identical whereas a metaphor is only partially similar to the target (Wozny, 1989). Critical to the power of metaphor is that the convocation of source and target ideas must involve some transformation, hence users do actively construct the relationships that comprise the metaphor during interaction with the system (Alty, Knott, Anderson, & Smyth, 2000). Salient dissimilarities of the ideas—in the context of salient similarities—stimulate thought and can facilitate active learning (Carroll & Mack, 1999). However, not all dissimilarities give rise to active learning and some may even hinder it. Therefore, the testing of metaphors is crucial.

Metaphor research approaches metaphors from three different angles: operational, structural and pragmatic (Carroll, Mack, & Kellogg, 1988). Operational approaches (e.g. Carroll & Mack, 1999; Mayer, 1975, 1976; Sajaniemi & Kuittinen, 2005) try to explain how metaphors operate in the mind to accomplish their effects; structural approaches (e.g. Alty et al., 2000; Gentner, 1983) focus on the structures of the target and the source in order to find some rules about the structural demands of a metaphor; and pragmatic approaches are empiric investigations about how metaphors work in practice. In order to design and implement systems with appropriate and functional metaphors all three approaches are essential. Some frameworks for this work have been suggested (Alty et al., 2000; Carroll et al., 1988; Madsen, 1994; Marcus, 1998), but we know of no report on simultaneous use of all three approaches in evaluating metaphors.

In order to test whether the role metaphors used in PlanAni reflect the properties of each role, please the users, and make their adopting of the role theory easier, we conducted a controlled experiment using all three approaches. This paper describes the experiment and discusses its results.

The rest of the paper is structured as follows. The next section contains a short introduction to roles and their use in program animation. Then, the experiment and its results are described and discussed. The paper is closed with a short conclusion.

Roles and Program Animation

Variables are not used in programs in a random or *ad-hoc* way but there are several standard use patterns that occur over and over again. In programming textbooks, two patterns are typically described: the counter and the temporary. Sajaniemi (2002) has generalized this idea to the concept of the roles of variables. His aim was to find a comprehensive, yet compact, set of characterizations of variables for the purposes of, e.g., teaching programming and analyzing large-scale programs.

A role characterizes the dynamic nature—or behavior—of a variable: the sequence of its successive values as related to other variables and external events. The way the value of a variable is used has no effect on the role. For example, a variable whose value does not change is considered

to be a fixed value whether it is used to limit the number of rounds in a loop or as a divisor in a single assignment. Furthermore, as roles describe behavior, they are related to the deep structure of programs and not to the surface structure, e.g., the form of assignment used to update a variable. Table 1 gives informal definitions of all roles identified in novice-level procedural programs; formal definitions can be found in the *Roles of Variables Home Page* at http://www.cs.joensuu.fi/~saja/var_roles/.

Table 1: Informal role definitions and role properties that should be visualized		
ROLE	INFORMAL DEFINITION	PROPERTIES
Fixed value	A variable which is initialized without any calculation and whose value does not change thereafter.	Impossible to change.
Stepper	A variable stepping through a succession of values that can be predicted as soon as the succession starts.	Future values can be predicted if past values are known; usually there is a direction for successive values: either upwards or downwards.
Most-recent holder	A variable holding the latest value encountered in going through a succession of values.	Successive values are obtained from some data series but they have no fixed relationship.
Most-wanted holder	A variable holding the "best" value encountered so far in going through a succession of values. There are no restrictions on how to measure the goodness of the value.	Current value is better than any of the previous values.
Gatherer	A variable accumulating the effect of individual values in going through a succession of values.	A new value is obtained by combining some new data and the previous value.
Transformation	A variable that always gets its new value from the same calculation from values(s) of other variable(s).	A new value is calculated from some other variables.
Follower	A variable that gets its values by following another variable.	Tightly connected to another variable; usually its previous value.
One-way flag	A two-valued variable that cannot get its initial value once its value has been changed.	Only two possible values; impossible to regain the initial value once changed.
Organizer	An array which is only used for rearranging its elements after initialization.	Individual parts cannot be changed but they can be moved around.
Temporary	A variable holding some value for a very short time only.	Exists for short time periods only.
Other	Any other variable.	No common properties.

Roles represent experts' tacit knowledge (Sajaniemi & Navarro Prieto, submitted) in a form that can be taught to novices. Role knowledge makes a difference in the position of variables within a program: in the traditional approach, a variable has no special meaning by itself but is only the object of some—to a novice apparently more or less incidental—assignments. Role knowledge turns this situation upside down: a variable is now an active subject taking care of some specific task and all assignments can be seen to support this task. Thus, roles make the deep program knowledge more accessible to students. Traditionally, students have had to acquire this kind of knowledge from example programs and program fragments. Introducing the roles explicitly gives students a comprehensive set of concepts and a vocabulary that they can use in studying example programs and in authoring new programs. Thus, roles are not a collection of additional concepts that enlarges the amount of material to be learned but they are an instrument for thinking. Students should not be graded on their ability to assign roles; instead, roles are design rules and pedagogical aids intended to help novices over the hurdle of learning programming.

Roles have been used in PlanAni (Sajaniemi & Kuittinen, 2003), a program animation system, which uses role images for variables and role-based animation for operations. In PlanAni, each role has a visualization—role image—that is used for all variables of the role. Role images give clues on how the successive values of the variable relate to each other and to other variables. For example, a fixed value is depicted by a stone giving the impression of a value that is not easy to change, and a follower as a dog positioned next to the variable whose value it follows. A stepper is depicted by footprints. It shows the current value and some of the values the variable has had or may have in the future. There is also an arrow giving the current direction of stepping. A gatherer is depicted as a box holding the current and the previous value, a temporary as a flashlight that is on just as long as the value is used, etc. The values of variables are superimposed on role images.

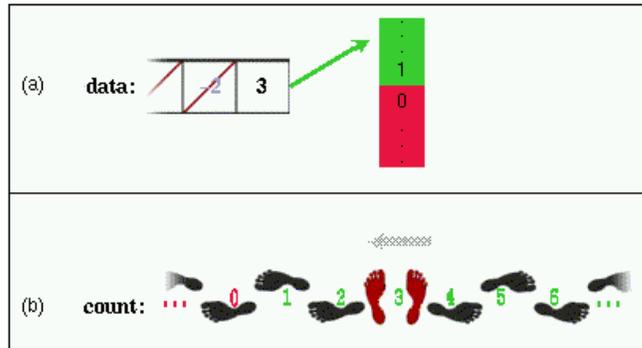


Figure 1: Visualizations of the same operation for different roles: comparing whether a most-recent holder (a) or a stepper (b) is positive.

Role images are intended to operate as visual metaphors that facilitate learning. According to Carroll and Mack (1999), metaphors stimulate thought processes and encourage students to use their existing knowledge thus improving their proficiency in novel situations. The third column of Table 1 presents the most essential properties of each role that were considered when designing metaphorical role images (Sajaniemi & Kuittinen, 2003).

In PlanAni the animation of operations depends on the role, also. For example, Figure 1 gives visualizations for the two syntactically similar comparisons “some_variable > 0”. In case (a), the variable is a most-recent holder and the comparison just checks whether the value is in the allowed range that appears on the screen when the comparison is made. In the visualization, the set of possible values emerges, allowed values with a green background and disallowed values with

red. The arrow that points to that part of the values where the current value of the variable lays, appears as green or red depending on the values it points to. Finally, the arrow flashes to indicate the result of the comparison. In case (b), the variable is a stepper with several values shown all the time within the role image. In this case, no new values appear; only the colors of the existing values change. The user can see the result by the color of the current value located in the middle of the footprints. If the current value is green, the value is in the allowed range, and if it is red, the comparison fails.

By using role images and role-based animation of operations, PlanAni visualizes roles instead of simply visualizing variables. Petre & Blackwell (1999) note that visualizations should not work in the programming language level because within-paradigm visualizations, i.e., those dealing with programming language constructs, are uninformative. Hence visualization of higher-level program constructs should be preferred to visualization of language-level constructs in teaching programming. PlanAni visualizes higher-level constructs, roles, which separates it from other program animators.

Figure 2 is an actual screenshot of the PlanAni user interface when the system is animating a simple program that checks whether its input is a palindrome. The left pane shows the animated program with a color enhancement pointing out the current action. The upper part of the right pane is reserved for variables, and below it there is the input/output area consisting of a paper for output and a plate for input. The currently active action in the program pane on the left is connected with an arrow to the corresponding variable on the right. Whenever the color enhancement is moved to a new location in the program, the new enhancement flashes to attract users' attention.

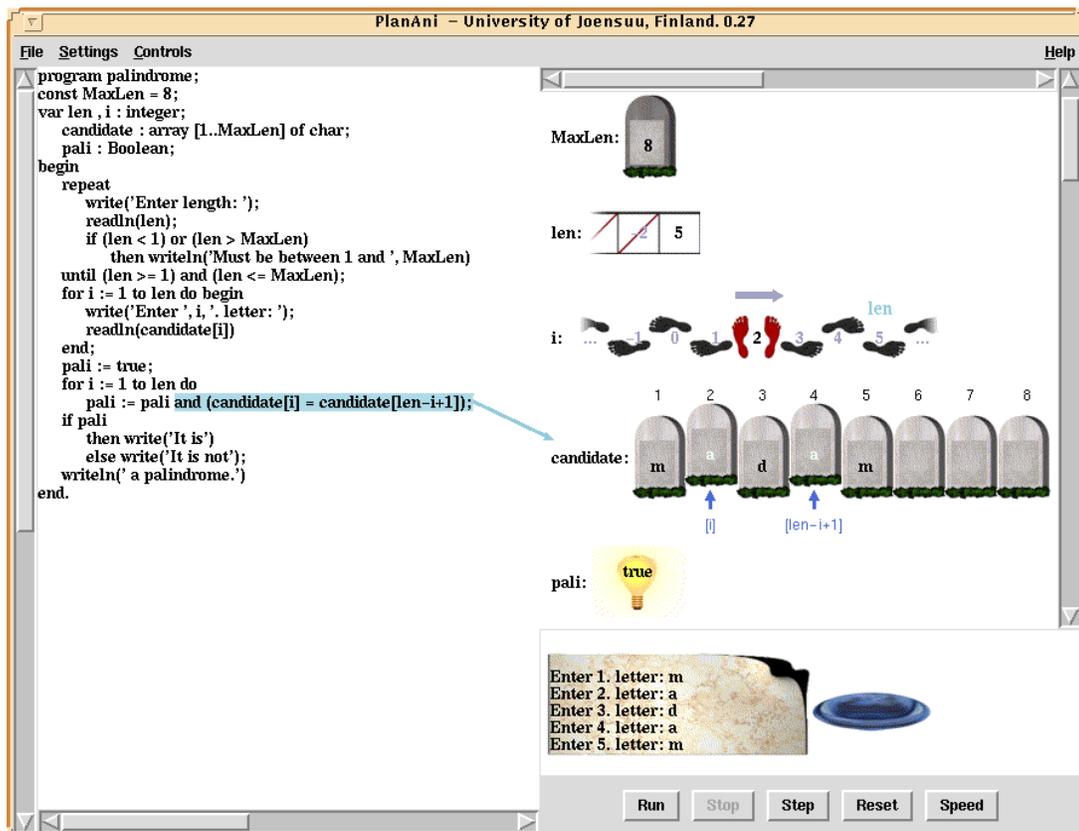


Figure 2: Visualization of an array element comparison in the PlanAni system.

During animation PlanAni uses frequent pop-ups that explain what is going on in the program. This includes variable creation (e.g., “*Creating a gatherer called 'sum'*”), operations (“*Testing whether the most-recent-holder 'data' is larger than 0*”), and control constructs (“*Entering a loop*”). Thus, role names appear continuously on the screen. To avoid unnecessary details PlanAni does not animate the evaluation of expressions: only the resulting value—accompanied by the expression itself—is shown and its effect in a comparison or assignment is animated.

PlanAni is implemented using Tcl/Tk and it has been tested both on Linux/Unix and Microsoft Windows. Even though the examples in this paper consist of Pascal programs, PlanAni is language independent. Example programs in Pascal, C, and Java are included in the distribution version freely available at the Roles of Variables Home Page.

Experiment

To test the hypothesis that PlanAni role images, understood as metaphors, reflect the special properties of roles, we conducted an experiment on five roles: stepper, fixed value, follower, gatherer, and temporary. The set of roles was limited in order to keep the subjects’ task manageable. The experiment was based on a framework for engineering user interface metaphors developed by Alty et al. (2000). For this purpose we created a set of alternative control metaphors (see Figures 3 and 4) using neutral images. Animation was controlled by using the same amount of animation effects for the control metaphor as for the corresponding PlanAni metaphor.

The control metaphors were deliberately chosen not to directly reflect role properties. The purpose of the experiment was not to find “optimal” role metaphors because individual preferences are different and it is doubtful that metaphors everybody would call perfect do exist. Instead, the purpose of the experiment was to evaluate whether PlanAni role images are acceptable metaphors in the sense that they facilitate learning. Had we chosen, in some sense, “good” control metaphors, variation of individual differences among subjects would have most probably been large when compared to the differences between the metaphors. The results would then have been statistically insignificant and no support for the positive effect on learning would have been discovered at all. By using neutral control metaphors, it was possible to reveal the effect of PlanAni metaphors.

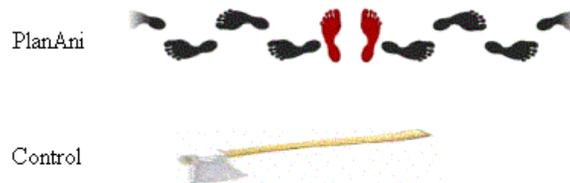


Figure 3: PlanAni and control metaphors for stepper.

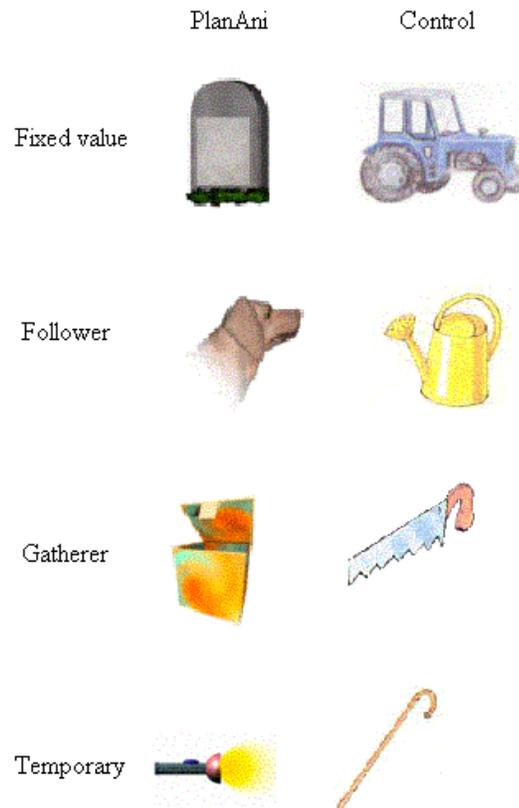


Figure 4: PlanAni and control metaphors for fixed value, follower, gatherer, and temporary.

In addition to finding whether PlanAni role metaphors facilitate learning, we were interested in finding deficiencies of individual metaphors. The set of control metaphors has no effect on this goal; thus neutral control metaphors were appropriate for the experiment.

Method

The experiment was a between-subject design with the metaphor set as the between-subject factor. All subjects had learned the role theory earlier but none had used the animator. The experiment session was started with a recap of the role theory where all ten roles were introduced and no images were used. The recap was followed by a pretest. The results of the pretest were used to divide the subjects into *PlanAni* and *control groups* that were equally good. In the experiment, the PlanAni group evaluated the original role metaphors and the control group the control metaphors. The experiment consisted of several phases: in the *first phase* the subjects reported their views of the images; in the *second phase* they used the PlanAni animator and their recognition of the metaphors after the use was tested; the *third phase* was a posttest on role knowledge.

Subjects

The subjects, thirteen second year computer science students, twelve male and one female, had earlier participated in a classroom experiment on variable roles (Sajaniemi & Kuittinen, 2005). The earlier experiment had used three groups of students instructed differently: in the traditional way with no specific treatment of roles; using roles throughout the course; and using a role-based

program animator in exercises in addition to using roles in teaching. The subjects of the current experiment came from the second group of the earlier experiment: they had had roles in teaching but they had not used PlanAni. Thus, the subjects had never seen role metaphors in action but had seen them once printed in black and white on paper.

Based on the pretest, seven subjects were assigned to the PlanAni group and six to the control group. The mean score of both groups was 6.7 (on scale 0–14).

Materials

In the recap, all subjects were given the same written material with descriptions of all roles and examples of their use. The pretest consisted of three small Pascal programs with 14 variables whose roles subjects had to recognize with the help of the material.

In the first phase, subjects' views of the images were collected using sheets of paper having the evaluated images—either role or control images—and space for writing verbs and adjectives associated with the images.

In the second phase, two versions of PlanAni were used, one with role metaphors and the other with control metaphors. A single program with variables representing all the evaluated roles was animated. Metaphor recognition was tested using a sheet having all just seen metaphors with space for answers, grades (using the Finnish school grading system 4-10 with 4 being the worst and 10 the best grade), and possible suggestions for “perfect” (i.e., “grade 10”) metaphors. The list of all roles with their informal definitions was printed at the bottom of the sheet.

The third phase—the posttest—consisted of three new Pascal programs with 14 variables whose roles subjects had to recognize.

Procedure

In the recap, subjects had 15 minutes to refamiliarize themselves with the role theory and to assign roles to the 14 variables of the pretest. This was followed by a short break; during the break the pretest was scored and the two subject groups formed.

In the first phase, subjects were given five minutes to write four associated adjectives for each of the five images, and another five minutes to give four verbs for each image. The images were printed on a paper and roles were not mentioned in any way; of course, the recap had primed role information.

In the second phase, subjects had ten minutes to watch the animation and make notes on changes in the values of the variables. They were then given ten minutes to write down what role was depicted by what metaphor, and to grade the metaphors and suggest alternative metaphors.

In the posttest, subjects had ten minutes to assign roles to variables in three new programs.

Results

The results provide information from three different angles: similarities and dissimilarities of the images and the associated roles; recognition and subjective grading of the metaphors; and the effect of metaphors on learning. In interpreting the results, statistical significance is based on the 5 % level.

Properties of Images

The verbs and adjectives given by subjects for the various images in the first phase were compared with essential properties of roles by three independent judges (all male, fourth to sixth year humanities students). The judges were given five word lists, one for each role. Each list contained

all verbs and adjectives given by all subjects to either the role or control image, sorted alphabetically. Each of the five lists was titled with few words describing essential properties of the role, e.g., for stepper: “predetermined, regular”. The judges were asked to mark words that had positive association with the title and, if possible, to mark approximately half of the words in each list.

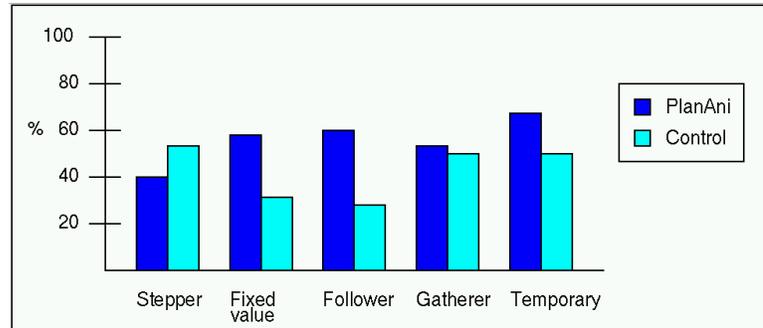


Figure 5: Proportion of role properties of images.

Figure 5 gives for each image the proportion of marked words of all words given to the image. For all other roles except stepper, the role images have a larger amount of correct associations than the control images. Based on chi squared test on average judge decisions, differences between groups are significant for the images representing fixed value ($\chi^2=7.083$, $df=1$, $p=0.0078$) and follower ($\chi^2=10.245$, $df=1$, $p=0.0014$) but not for other roles.

Subjects gave for the PlanAni role images also verbs and adjectives that were opposite to actual role properties. Most notable suggestions were given to stepper, with 26 % of properties relating to uncertainty and 11 % relating to argument or separation, and gatherer, with 10 % relating to unclarity and ambiguity.

Metaphor Recognition and Grading

The second phase tested how subjects recognized roles from the metaphors after seeing them during PlanAni use. Figure 6 presents the proportion of subjects recognizing each metaphor. The PlanAni 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 ($p=0.0862$).

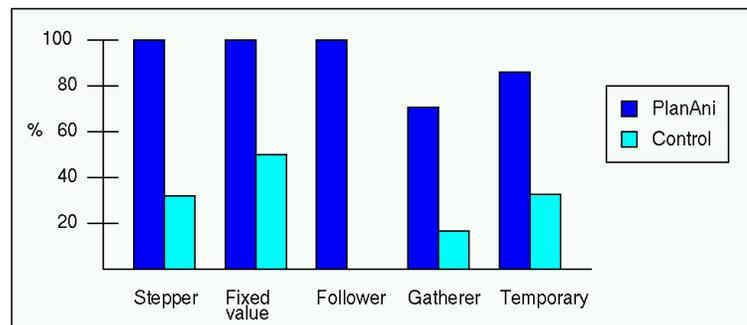


Figure 6: Proportion of metaphor recognition.

The subjects also suggested alternative metaphors for roles, especially the control group was eager to propose better role metaphors. For *stepper* a walking man, a staircase and a myriapod was suggested. For *fixed value* a rock, a stone, a pyramid, a tree and the sun were suggested. The al-

ternatives for *follower* were a shadow, a child following his mother, a tracking dog and a man following somebody’s footsteps. The suggestions for *gatherer* were pieces of a puzzle, a meeting place, a trash can, a construction crane, a basket and a box containing something. For *temporary* a plastic bag, a bird cage, some goods transferring from one shelf to another, a locker in supermarket, a toilet-seat, a watering can, a glass, a hat turned downwards and a paper cup were suggested.

Table 2 presents means of grades that the subjects gave to the metaphors. These figures include only grades by subjects 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 subjects of the control group recognized these metaphors correctly.

Table 2: Means of role metaphor grades (scale 4-10; 10 being best)

GROUP	STEPPER	FIXED VALUE	FOLLOWER	GATHERER	TEMPORARY
PlanAni	9.3	8.9	8.0	8.8	5.8
Control	5.5	6.7		5.0	5.0

Effects on Learning

The experiment started with a recap of roles and a pretest where subjects assigned roles to 14 variables. The experimental groups were formed based on this pretest so that the two groups became equally good. At the end of the experiment, a posttest was conducted. Again, the subjects assigned roles to 14 new variables. The mean number of correctly recognized roles in the pretest and in the posttest are given in Table 3. The last column gives the mean changes from pretest to posttest for both groups. Based on Wilcoxon’s rank sum test, the difference of change between the groups is significant ($p=0.0012$).

Table 3: Performance of the groups in the pretest and posttest (scale 0-14; 14 being best)

GROUP	PRETEST	POSTTEST	CHANGE
PlanAni	6.7	11.4	4.7
Control	6.7	5.8	-0.9

Discussion

The posttest represents the operational approach to metaphor research, and the analysis of image properties represents the structural approach. The rest of the results, metaphor recognition and grading, represent the pragmatic approach. We have used the operational approach to see the overall effect of metaphors functioning together, the pragmatic approach to compare the quality of individual metaphors and the structural approach to find out specific problems within each metaphor.

The materials in the pretest and posttest were different making a direct comparison of their results impossible. However, the difference between the groups is clear and statistically significant ($p=0.0012$). Use of the original role metaphors in the animator and in other tasks of the experiment facilitated learning when compared with the use of control metaphors. This finding motivates the whole study: the selection of metaphors makes a measurable difference in learning out-

come. This makes the evaluation and improvement of individual role metaphors a sensible task to do.

The subjects gave information about each image both before they knew about its connection to the role theory (phase 1) and after they had seen the metaphor in action in PlanAni (phase 2). These different kinds of measures will now be combined to get a versatile view of the quality of the individual metaphors.

The role image for *stepper* reflects the properties of the role little worse than the control image and 26 % of its properties did associate with uncertainty and 11 % with argument. However, it is recognized better than the control metaphor ($p=0.0210$) and in subjects' opinion also represents the role better ($p=0.0014$). The image properties were collected before the PlanAni session where the animation of assignment operations clearly demonstrated the predictability of the value sequence. This probably made recognition easier and explains the contradiction between poor properties of the image and good recognition of the metaphor. Moreover, the image—footprints—has a clear connection with the verbal metaphor of the role, i.e., its name (*stepper*). The perfect recognition and highest grade are probably affected by this double metaphor effect. However, the high percentage of unwanted properties suggests that the image should be elaborated so that it would not give the impression of uncertainty or argument.

The role image for *fixed value* reflects the properties of the role better than the control image ($p=0.0078$), is recognized perfectly even though the control metaphor did well too, and got better grades than the control metaphor ($p=0.0196$). Alternatives suggested for the control metaphor included rock and stone; thus the role image seems to be an appropriate metaphor.

The role image for *follower* reflects the role clearly better than the control image ($p=0.0014$), and is clearly more recognizable ($p=0.0006$). The role metaphor also obtained a good grade which could not be compared with the control metaphor because no subject in the control group recognized the watering can as a metaphor for follower. This suggests that the role metaphor is appropriate for its task.

The role image for *gatherer* reflects the properties of the role as poorly as the control image, but is better recognized ($p=0.0775$), and got a very good grade. This seems to suggest that watching the animation enhanced the still image with other information. The animation demonstrated how the new values of the variable were obtained and thus made the role characteristics visible. However, this metaphor had the lowest recognition frequency of all role metaphors (even though better than any of the control metaphors) and was often confused with the role temporary. This metaphor certainly needs improvements.

The role image for *temporary* does not reflect the role properties any better than the control image, but it was slightly better recognized ($p=0.0862$). This could be explained with the fact that in PlanAni the flashlight turns on when the variable gets a new value, and off when the value has been used and is not needed anymore. This behavior reflects the essential property of the role temporary and may support recognition. The grade was very low and it is clear that the metaphor must be totally redesigned.

Table 4 gives a summary of the numeric criteria used to compare the role and control metaphors. Alty et al. (2000) note that the amount of metaphor's properties reflecting the idea, i.e., role in this case, is essential in evaluating the goodness of a metaphor. This is demonstrated in the table by the roles fixed value and follower, for which good properties imply good recognition and grade. An evaluation based on properties is easy to accomplish: subjects need no knowledge of the idea the metaphor is about. Thus, anybody can be used as an evaluator. However, for *stepper* the properties of the image were poor even though the metaphor was well recognized. For this role, the animation provided essential information that was not readily available in the still pic-

ture. We suggest that in such cases the evaluation of a metaphor should be based on actual use even though this means that the animation must be implemented and subjects need to understand the ideas that the metaphors represent.

There are some threats to the validity of the experiment. The subjects had earlier seen the role images printed on paper. Moreover, they had studied programming for almost two years whereas the role images are directed to novices. As a result there might be a positive effect on role image recognition. On the other hand, the role images were utilized in their normal setting and the subjects had the appropriate knowledge of roles. Thus, they could give rational comments on the quality of the metaphors.

The evaluation of image properties was based on judges that were given only short descriptions of the essential properties of roles. The judges were unaware of programming concepts and of the role theory. On the other hand, the judges were humanities students and therefore able to interpret the tones of words well. Moreover, the evaluation of the metaphors was versatile, and the evaluation of properties was only a part of it.

Table 4: Differences between groups (* $p = 0.05-0.10$, ** $p = 0.01-0.05$, * $p < 0.01$)**

CRITERION	STEPPER	FIXED	FOLLOWER	GATHERER	TEMPORARY
Properties		***	***		
Recognition	**	*	***	*	*
Grade	***	**	N.A.	N.A.	

Conclusion

We have reported the results of an experiment evaluating the quality of role images used as visual metaphors in the PlanAni program animator. The experiment compared role metaphors with control metaphors using several evaluation criteria: properties of the images, metaphor recognition and grading, and effects on learning. These criteria represent the three approaches to metaphors: structural, pragmatic, and operational. The results demonstrated that as a whole the visual role metaphors facilitate learning and identified ideas for further elaboration of the individual metaphors.

The control metaphors were deliberately chosen not to directly reflect role properties. The purpose of the experiment was not to find “optimal” role metaphors because individual preferences are different and it is doubtful that metaphors that everybody would call perfect do exist. Instead, the purpose of the experiment was to evaluate whether PlanAni role images are acceptable in the sense that they facilitate learning, and to find deficiencies of individual metaphors. By using neutral control metaphors both of these tasks could be accomplished.

The experiment gave also light on visual metaphor evaluation methods. An evaluation based on properties is easy to accomplish because it poses no special requirements on subjects. However, in cases where animation provides additional content to a metaphor, a more laborious evaluation may be needed demanding subjects to know the ideas more thoroughly.

Acknowledgments

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

References

- Alty, J. L., Knott, R. P., Anderson, B., & Smyth, M. (2000). A framework for engineering metaphor at the user interface. *Interacting with Computers*, *13*, 301–322.
- Ben-Ari, M. & Sajaniemi, J. (2004). Roles of variables from the perspective of computer science educators. In *The 9th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2004)*, pp. 52–56. Association for Computing Machinery.
- Carroll, J. M. & Mack, R. L. (1999). Metaphor, computing systems, and active learning. *International Journal of Human-Computer Studies*, *51*, 385–403.
- Carroll, J. M., Mack, R. L., & Kellogg, W. A. (1988). Interface metaphors and user interface design. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 67-85). Amsterdam: Elsevier Science Publishers, B.V.
- Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*, 155-170.
- Kuittinen, M. & Sajaniemi, J. (2004). Teaching roles of variables in elementary programming courses. In *The 9th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2004)*, pp. 57–61. Association for Computing Machinery.
- Madsen, K. H. (1994). A guide to metaphorical design. *Communications of the ACM*, *37*, 57-62.
- Marcus, A. (1998). Metaphor design in user interfaces. *Journal of Computer Documentation*, *22*, 43-57.
- Mayer, R. E. (1975). Different problem-solving competencies established in learning computer programming with and without meaningful models. *Journal of Educational Psychology*, *67*, 725-734.
- Mayer, R. E. (1976). Some conditions of meaningful learning for computer programming: advance organizers and subject control of frame order. *Journal of Educational Psychology*, *68*, 143-150.
- Ortony, A., (Ed.). (1993). *Metaphor and Thought* (2nd ed.). Cambridge University Press.
- Petre, M., & Blackwell, A. F. (1999). Mental imagery in program design and visual programming. *International Journal of Human-Computer Studies*, *51*, 7-30.
- Sajaniemi, J. (2002). An empirical analysis of roles of variables in novice-level procedural programs. In *Proceedings of IEEE 2002 Symposia on Human Centric Computing Languages and Environments (HCC'02)*, pp. 37–39. IEEE Computer Society.
- Sajaniemi, J. & Kuittinen, M. (2003). Program animation based on the roles of variables. In *ACM 2003 Symposium on Software Visualization (SoftVis 2003)*, pp. 7–16. Association for Computing Machinery.
- Sajaniemi, J. & Kuittinen, M. (2005). An experiment on using roles of variables in teaching introductory programming. *Computer Science Education*, *15* (1), 59-82.
- Sajaniemi, J. & Navarro Prieto, R. (submitted) Roles of variables in experts' programming knowledge. Submitted to the *17th Annual Workshop of the Psychology of Programming Interest Group*.
- Wozny, L. A. (1989). The application of metaphor, analogy, and conceptual models in computer systems. *Interacting with Computers*, *1*, 273–283.

Biographies



Tuija Stütze received the Designers degree in computer science in 1982 from the Institute of ADP in Helsinki, Finland and the Masters degree in computer science in 2005 from the University of Joensuu, Finland. Since 1983 she has been working in several software houses as programmer, designer, systems analyst and project manager.



Jorma Sajaniemi received the Licentiate degree in computer science in 1975 from the University of Helsinki, Finland. Since 1979, he has been associated with the Department of Computer Science at the University of Joensuu, Finland, as an associate professor and a full professor. He has obtained industrial experience in Softplan Ltd and Karjalan Tietovalta Ltd. His main area of research is psychology of programming and he has focused on mental models and cognition-based tool support in programming.