**Social Science Computer Review**
Volume 25 Number 3
Fall 2007 405–409

10.1177/0894439307297693
http://ssc.sagepub.com
hosted at
http://online.sagepub.com

# Assessing Relationship Closeness Online

## Moving From an Interval-Scaled to Continuous Measure of Including Others in the Self

Benjamin Le
William B. Moss
*Haverford College, Haverford, Pennsylvania*
Debra Mashek
*Harvey Mudd College, Claremont, California*

A continuous measure of relationship closeness inspired by the Inclusion of Other in the Self Scale was designed using Java. This new measure allows for closeness to be assessed on a continuous scale of zero to 100, with output values corresponding to the degree of overlap and distance between the objects in the applet. In addition, the applet includes options to enhance its flexibility and usefulness in research applications. In particular, the behavior of the applet, and properties of objects included in it, can be customized. The construction of the applet is described, and methodological and theoretical considerations regarding this new measure are discussed.

***Keywords:*** *measurement; scaling; closeness; inclusion; applet; methodology; programming*

Close relationship researchers face the challenge of how to best measure relational variables such as closeness, love, commitment, and satisfaction (cf. Berscheid, Snyder, & Omoto, 1989; Hatfield & Sprecher, 1986; Rusbult, Martz, & Agnew, 1998). Closeness, in particular, is an important variable in relationship research, and Aron and colleagues (Aron, Mashek, & Aron, 2004) have argued that closeness can be described as a holistic process of cognitively including another person within one's self-concept. Consistent with this approach, they use the analogy of overlapping circles (i.e., a Venn diagram) representing the self and other to illustrate this theoretical foundation for understanding closeness. Based on this model of closeness, Aron and colleagues (Aron, Aron, & Smollan, 1992) constructed the Inclusion of Other in the Self Scale (IOS) as a measure of dyadic closeness. The IOS is a series of seven Venn-like diagrams, each depicting two circles in various states of overlap from not overlapped at all to nearly completely overlapped. The circles are labeled as *self* and *other*, and typically a research participant is instructed to select the set of circles that best represents his or her current relationship with another person, such as his or her romantic partner. This theoretical approach for understanding and measuring closeness has garnered much support (Agnew, Loving, Le, & Goodfriend, 2004) and has the advantage of relying on a short, single item that can quickly and easily be used to assess closeness.

In this report, we describe the construction of a dynamic, continuously scaled adaptation of the IOS in the form of a Java applet designed to be embedded within a Web page. In addition, we offer a brief tutorial on using the applet, including a description of the enhancements and extensions that the applet offers beyond the traditional paper-and-pencil IOS. Information about the applet, including the code, full documentation, a demonstration, and examples, is available from the first author's Web page (`http://www.haverford.edu/psych/ble/continuous_ios`) (Le & Moss, 2007).

## Development of the Continuous IOS

Because a dynamic and continuous version of the IOS requires the ability to overlap circles, a technology more robust than standard HTML was needed to allow the measure to be embedded within a Web-based survey. Therefore, we employed a Java applet and JavaScript to interface the applet with the rest of the Web page. This allows for multiple copies of the applet to be embedded in a single Web survey along with other standard HTML form elements. In addition, using JavaScript allows data to be returned from the applet in the same manner as a standard HTML form element, with only a few minor modifications to the existing HTML required.

The applet was written using the JFC/Swing or JApplet architecture provided in the standard Java development package. This requires the user (i.e., the research participant) to have a newer version of Java than the regular applet architecture; however, using JApplet made sense both currently, given it provides more features than the standard applet toolkit, and in the long term, because future enhancements to the Java development package will be made for JApplet.

The JApplet architecture provides a comprehensive toolbox of methods that make drawing and writing to the screen simple matters of calling the correct functions. When the user clicks and drags the mouse to overlap the circles (indicating the amount of closeness in his or her relationship), the clicking and dragging are handled by the applet interface, and writing the code was simply a matter of implementing the necessary functions from the interface (i.e., mousePressed and mouseDragged). The Java interface provides the location of the user's click to the functions, so they need only to ascertain whether the location of the user's click is within the movable circle and, if so, adjust the stored location of the circle. Drawing the new location of a circle to the screen requires only two lines of code—one to set the color and another to draw the circle. The labels on the circles, which can be changed from *self* and *other* (i.e., the standard IOS labels) by the survey designer, are then drawn onto the circles. The applet currently supports three methods for drawing the labels: manual, auto, and the default. The default behavior simply draws the text for the label onto the circle, adjusting the location of the text so it fits within the bounds of the circle. The manual mode allows the user to have multiple lines in the text that is being drawn, separated by the <br> tag, like in standard HTML. Finally, the automatic mode attempts to break up the label into the most efficient number of lines, and then draws those onto the circle. Unlike other functionality used in the applet, adjusting the location of the labels so they fit within the circles is not provided by the JApplet framework. This feature, therefore, necessitated the most complicated and lengthy code in the applet. The width of the text is calculated using the Java text library. The proper distance to place the text from the center of the circle is then calculated using the text width and the radius of the circle. In the automatic mode, the text is split into lines using an implementation of the greedy algorithmic technique and then drawn to the screen in much the same way. Finally, when the form is submitted, the value for the overlap of the two circles is calculated using a standard circle-circle intersection function, and the distance is found and adjusted to place the value between 0 and 100.

## Using the Applet

Researchers using the Continuous IOS in a Web-based survey must add a few simple elements to their Web pages for the applet to function correctly. First, the applet needs to be added to the Web page. This is done using the <applet> html tag. The applet tag requires four options to function correctly: *height*, *width*, *name*, and *codebase*. The first two tags act as expected; they set the height and width of the applet area within the Web page. Next, *name* is simply a unique name for this instance of the applet (i.e., "IOS1"), and *codebase* tells the Web browser where to find the applet. With the addition of just the applet tag, the slider will appear in the Web page and be fully operational. A few other tags, however, are required to collect the data returned by the applet using a standard HTML form. First, two hidden form elements that will temporarily hold the data when the form is submitted must be added. These elements can be given any unique name, just like any other elements in the form. Second, a simple JavaScript function is needed. This function takes the values from the applet and stores them in the form elements that were just created; this allows the survey designer to use the same data collection methods for the data from the applet as he or she is using for the rest of the regular HTML fields. This function is practically the same for every Web page and can be copied almost directly from the applet documentation.

Finally, there are a number of options that can be set within the applet tag to further customize the applet. The radius of the circles will default to one half the height of the applet, making the circles as large as possible within the applet area. The user, however, may modify this value when creating the Web page. Furthermore, by default, the applet will load with the two circles starting directly adjacent to one another (i.e., akin to the standard IOS) at the rightmost edge of the applet. If a different starting location is desired, the starting locations of the two circles may be specified. The applet also allows for the appearance of the two circles to be customized to the user's specifications. The colors of the two circles may be specified, as well as the style, color, size, and display text for the labels for the circles. In particular, the blending of the colors as the circles overlap enhances the metaphor of self-other inclusion, and the ability to customize the labels makes the measure flexible for many research applications assessing closeness between entities besides self and other. The applet will attempt to keep the label within the confines of the circle by adjusting the location and layout of the text; however, if a single word is larger than the diameter of the circle, there is little the applet can do. Finally, the user can configure whether or not the left circle can move towards and away from the right circle, or only towards it. This option may have both theoretical and practical implications for the notion of closeness as a dimension of overlap between individuals or as an indicant of physical distance between individuals. The specific format for specifying the options is available in the full documentation for the applet.

## Outputted Data Format

As previously mentioned, data from the Continuous IOS applet are outputted and can be collected along with responses from other form objects (e.g., radio buttons, text fields, and menus). Specifically, two values are collected from the applet. First, and most consistent with theory (Aron et al., 2004), the amount of overlap between the two circles is outputted as a percentage between 0% (i.e., no overlap) and 100% (i.e., complete overlap between the two circles). In addition, the distance that the left circle has moved is assessed. More specifically, this distance is assessed on a scale of zero to 100 units, with 100 units representing the diameter of the circle. In other words, the default position of adjacent circles is a distance of zero; moving the left circle all the way to the right (i.e., complete overlap) represents a distance of 100. If the applet is set to allow the left circle to move away from the right circle, negative distance is outputted. Note that the overlap and distance values are very highly correlated with one another, but both are outputted because they represent distinct theoretical dimensions and may have different ranges. Specifically, overlap is theoretically consistent with the inclusion of other in the self model and is most analogous to the standard IOS, although with the benefit of added sensitivity of a zero- to 100-point scale (rather than the standard 7-point scale). Configuring the applet to allow the left circle to move away (i.e., to the left) allows for pulling the self away from the other (i.e., negative distance). In this case, distance would return a negative value, whereas overlap would not be sensitive to varying degrees of negative distance (i.e., it would always be 0% for nonoverlapping circles, no matter the distance between them).

## Conclusions

Methodologically, the Continuous IOS allows for Web-based usage of the IOS, which is useful in gathering data from large samples of participants. In addition, the options available for customizing the applet make it attractive. For example, the *other* can be precisely noted with specific labels such as *my best friend* or *my dating partner*. Furthermore, it is possible that the increased measurement precision and variability associated with the Continuous IOS will strengthen the predictive power of closeness relative to other relational variables.

From a theoretical perspective, the Continuous IOS offers exciting possibilities for testing hypotheses relating to the foundations of closeness. For example, the ability to assess both distance (e.g., movement away from the other) and amount of overlap would allow researchers to investigate if closeness is in fact a state of self-other inclusion, or if it is a function of physical and psychological distance. In addition, measures such as the Continuous IOS may allow for a more dynamic understanding of dyadic processes associated with closeness such as self-expansion (e.g., Lewandowski, Aron, Bassis, & Kunak, 2006). Although driven by technological advances, a worthwhile measure must also facilitate theoretical advances in its given field. It is our hope that the Continuous IOS can serve this function in the study of close relationship processes.

## References

Agnew, C. R., Loving, T. J., Le, B., & Goodfriend, W. (2004). Thinking close: Measuring relational closeness as perceived self-other inclusion. In D. Mashek & A. Aron (Eds.), *Handbook of closeness and intimacy* (pp. 103–115). Mahwah, NJ: Lawrence Erlbaum.

Aron, A., Aron, E. N., & Smollan, D. (1992). Inclusion of other in the self scale and the structure of interpersonal closeness. *Journal of Personality and Social Psychology*, *63*, 596–612.

Aron, A., Mashek, D., & Aron, E. (2004). Closeness as including other in the self. In D. Mashek & A. Aron (Eds.), *Handbook of closeness and intimacy* (pp. 27–41). Mahwah, NJ: Lawrence Erlbaum.

Berscheid, E., Snyder, M., & Omoto, A. M. (1989). The relationship closeness inventory: Assessing the closeness of interpersonal relationships. *Journal of Personality and Social Psychology*, *57*, 792–807.

Hatfield, E., & Sprecher, S. (1986). Measuring passionate love in intimate relationships. *Journal of Adolescence*, *9*, 383–410.

Lewandowski, G. W., Aron, A., Bassis, S., & Kunak, J. (2006). Losing a self-expanding relationship: Implications for the self-concept. *Personal Relationships*, *13*, 317–331.

Le, B., & Moss, W. B. (2007). About the Continuous IOS. Retrived March 30, 2007 from Haverford College, Department of Psychology: http://www.haverford.edu/psych/ble/continuous_ios/

Rusbult, C. E., Martz, J. M., & Agnew, C. R. (1998). The investment model scale: Measuring commitment level, satisfaction level, quality of alternatives, and investment size. *Personal Relationships*, *5*, 357–391.

**Benjamin Le** is currently an assistant professor of psychology at Haverford College, where he teaches classes in social psychology, close relationships, research methods, and statistics. His research focuses on the role of commitment, closeness, cognition, and emotion in close relationship processes. He can be contacted by e-mail at ble@haverford.edu or by mail at Haverford College, 370 Lancaster Ave., Haverford, PA 19041.

**William Moss** graduated from Haverford College in 2005 with a B.S. in computer science and physics. He currently lives in Brooklyn, works for a software company in Manhattan, and is applying to Ph.D. programs in computer science. He can be contacted by e-mail at wmoss@alum.haverford.edu.

**Debra Mashek** is an assistant professor of psychology at Harvey Mudd College, the liberal arts school of science and engineering. Her research examines interpersonal connectedness at the level of both dyads and communities. She can be contacted by e-mail at mashek@hmc.edu.