Focus in Graphical User Interfaces

Manuel A. Pérez

Human-Computer Interaction Lab
Naval Research Laboratory
Washington DC 20375-5320
perez@itd.nrl.navy.mil

John L. Sibert

Department of Electrical Engineering and Computer Science
The George Washington University
Washington DC
sibert@seas.gwu.edu

ABSTRACT
Focus in natural language processing is used to keep track of the attentional space of the participants in the dialogue. Graphical interfaces have still to benefit from the use of this level of communication. We discuss a graphical interface counter part called a focus space. Focus spaces can be used to combine multiple interaction styles (e.g. natural language with graphical pointing). We describe how a specific theory of discourse structure and focus space can be applied to graphical interfaces. The attentional space is maintained automatically by the system, and can be used by the software under program control. An example is provided showing how the attentional space can be used.

KEYWORDS: Focus, Discourse Structure, Graphical User Interfaces

INTRODUCTION

Discourse structure is an important aspect of natural communication. In natural communication, discourse structure (DS) can be used to resolve referring expressions (definite nouns, anaphora, pronouns), to resolve deixis (pointing references such as this, that, there, etc.), to indicate the intentions or goals of the speaker, and to generate referring expressions and deixis as well. The study of DS has been centered mostly around verbal communication.

Designers of user interfaces can benefit from the use of the theories of DS representation and communication. These theories can serve as the foundation for new software architectures that facilitate: generation of context sensitive messages, generation and interpretation of referring expressions and deixis, combination of modalities in the generation of system responses (e.g. graphics and text for help purposes), and understanding of multiple input devices combined for one utterance (e.g. natural language and pointing).

Focus represent the attentional space of a participant in a dialogue. Natural language systems use attentional space to track the current focus of the dialogue and to resolve ambiguous utterances. Direct manipulation interfaces can also benefit from the use of a focus representation. The equivalent of a focus representation has been used in limited situations in graphical interfaces, but only in an ad-hoc way. User interface building tools and their corresponding software architectures do not provide for the representation and management of focus.

An example of a graphical editor will show how focus spaces can be maintained based on user interaction with the application.

FOCUS SPACES

Discourse structure is decomposed into related parts: linguistic structure, intentional structure and attentional space [3]. The application of attentional space to user interfaces will be the focus of this work. Intentional structure is being studied by researchers doing planning recognition [1], and falls outside the scope of our work.

Focus partitioning is an organization of the concepts in the domain of a dialogue to help in the interpretation and understanding of utterances. It represents the context in which an utterance must be interpreted. In a natural language processing system, attentional spaces help to solve many of the ill-formed utterances inherent in natural languages. Graphical interfaces are designed without many of these characteristics, thus a representation for focus has not been used.

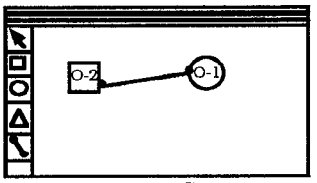

Figure 1. Sample Application

Two situations in which a focus space has been used indirectly in graphical interfaces are: "focusing" of keyboard and selection sets. "Focusing" of the keyboard refers to the selection of the interface widget that will receive input from the keyboard. Which widget is used is determined by simple rules. In some systems, the user must explicitly click on the widget that will receive the input. Other systems simply require the mouse to be within the area on the screen occupied by the widget. Yet others require the window containing the widget to be at the top of all other windows. In all of these cases, the mouse is being used to select the appropriate widget. The appropriate widget is being selected as the item in focus.

The idea of a current selection set is another example of a focus space used in graphical interfaces. The current selection is used in graphical applications to have one common way to select objects for actions. When actions are invoked, they take their arguments from the currently selected set of objects. The selection set becomes the focus of the dialogue.

Focus spaces in natural language conversations are much more complex than those described above. They divide the domain of discourse in hierarchical spaces organized according to the structure of the discourse. The focus spaces described here are a simplification of the theory of discourse structure of Barbara J. Grosz [2]. The domain of the discourse is partitioned into spaces. This partitioning is called focus partitioning. The partitioning reflects the structure of the discourse as it progresses. There is at most one active focus space at any time, which represents the concept that is the current focus of the dialogue. Previously active spaces are considered opened. Focus spaces can be explicit or implicit. Explicitly focused spaces are those that have been referenced directly in the dialogue. Implicitly focused spaces include information related to items that are in explicit focus.

EXAMPLE

In this section, we present a simple application that will serve to show the use of focus spaces in graphical interfaces. The application used is a graphical editor with four types of objects: squares, triangles, circles and connections. The first three types have two attributes associated with them: location and line type. The semantic meaning of the four types is left undefined. The purpose here is to show a graphical application typical of direct manipulation systems. Figure 1. shows the main window of our sample application. It has one square (O-2) and one circle (O-1). The square and the circle have been attached by a connection.

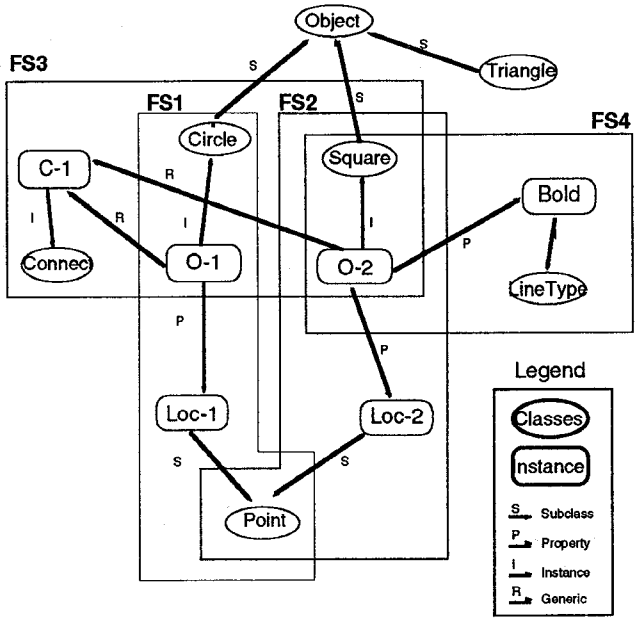

Figure 2. Knowledge Representation for Sample Domain

A knowledge representation is an important component for the use of a focus space. In our setting, the knowledge representation describes the application components of the program. The knowledge representation of the sample application is depicted in Figure 2 using a semantic network. This network contains nodes and arrows with labels to establish relationships among objects in the domain. The next section describes the focus spaces in this diagram.

FOCUS SPACE REPRESENTATION

The Figure 2 contains four explicit focus spaces[1], labeled FS1 through FS4. These spaces were created as the interaction progressed. The following is the sequence of actions performed at the user interface.

Action	Create FS	Active FS after action
Create O-1	FS1	FS1
Create O-2	FS2	FS2
Connect O-1 & O-2	FS3	FS3
Move O-1	-	FS1
Set Line Type O-2	FS4	FS4

[1] In this example, we use only the explicit focus spaces. Due to the small and simple domain, the implicit focus spaces are almost nonexistent.

FS1 is the active space after the first action is performed. It includes the object involved in the operation (O-1), its type information (circle), and any attributes that are set as the result of the action (Loc-1). The type information of the attributes is also included. The second and third action (create O-2 and connect O-1 to O-2) follow a similar pattern to the one in the first action. The result of each of the three actions is to create a new focus space and leave that space as the active space. After the move operation is performed, no new focus space is needed. All objects referenced in the action are part of the FS1 and thus, FS1 is activated. The last action, set line type, makes references to attributes of O-2 that were not in any focus space. Thus, a new focus space is created.

Note in Figure 2 that O-2 belongs to three different focus spaces. In FS1, O-2 is one of the attributes to the connection C-1. In FS2, O-2 is described with all its attributes. In FS4, O-2 is modified by the set line type operation. This "differential access" [2] is one of the advantages of focus spaces. Objects or concepts from the domain can be looked at from different perspectives according to how they are used in each utterance.

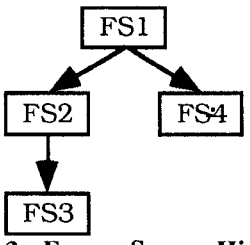

Figure 3. Focus Space Hierarchy

As the focus spaces are created, they are organized in a hierarchy that shows the structure of the discourse. The organization (shown in Figure 3 but not in Figure 4) takes the form of a tree, where the parent child relationship indicates a focus shift from parent to child.

USE OF FOCUS SPACES

Focus spaces can be used for several purposes, as the natural language processing community has shown. In our example, the focus alone doesn't provide much extra advantage to the graphical interfaces in use. The real advantage comes when we extend today's interfaces with multi-modal input (voice and pointing). Another example is an extension to the idea of the current selection set.

If natural language (keyword) input is available to the user, the focus space that was created and maintained by the graphical interface can be used by the natural language processing system to solve ill-formed utterances, just like it is used today in purely linguistic processing. The same economy of expression can be used by the system for output. For example, generating messages for the user, the system can use the focus space. References to objects not in focus need to be qualified fully (e.g. the connection between object X and Y), meanwhile references to objects in focus can be more concise (e.g. the connection).

The notion of the current selection set can be enhanced by the use of focus. For example, in our application some commands might apply only to connections and not to the other objects. Those commands are disabled when the selected set contains objects other than connections. This requires the user to create a selection set with only those items that are valid for the action. In complex applications, managing the selection set becomes a complex task on its own. The focus space can be used to complement the selection mechanism used. For example, if an item of certain type (e.g. connections) is available in the focus space and no items are selected, then execution of a command may be able to use the items in the focus space as implicit parameters.

ACKNOWLEDGEMENT

The authors will like to thank the Dialogue Project discussion group at the HCI Lab for the stimulating discussions that helped form many of the ideas presented here.

CONCLUSIONS

Focus spaces, as defined in the computational linguistic literature, keep track of the attention of the participants in a dialogue. They have been used in natural language processing to resolve ill-formed referents inherent in natural language. We have presented here how focus and a representation of focus space can be used in graphical interfaces. Two examples were described of possible uses for focus spaces in future graphical interfaces.

BIBLIOGRAPHY

1. Cohen, P.R., J. Morgan, and M.E. Pollack, ed. *Intentions in Communications.* System Development Foundation, 1990, The MIT Press: Cambridge, MA. 508.

2. Grosz, B.J., *The Representation and Use of Focus in a System for Understanding Dialogues,* in *Readings in Natural Language Processing,* B.J. Grosz, K.S. Jones, andB.L. Webber, Editor. 1986, Morgan Kaufmann Publishers, Inc.: Los Altos, California. p. 353-362.

3. Grosz, B.J. and C.J. Sidner, *Attentions, Intentions, and the Structure of Discourse.* Computational Linguistics, 1986. 12(3): p. 175-204.