Appeared in Proceedings of the International Symposium on Future Software Technology (ISFST-99), Nanjing, China, Software Engineers Associates, pp.113-118, October, 1999.

EVIDII: An Environment for Constructing Shared Understanding Through Visualizing Differences of Impressions

Masao Ohira[1], Yasuhiro Yamamoto[1], Kumiyo Nakakoji[1,2,3]

[1]Graduate School of Information Science, Nara Institute of Science and Technology
8916-5, Takayama-cho, Ikoma, Nara, 630-0101, Japan
{masao-o, yasuhi-y, kumiyo}@is.aist-nara.ac.jp
[2]Software Engineering Lab., SRA Inc.
[3]PRESTO, JST

ABSTRACT

This paper presents our approach to develop a computer system that supports cross-cultural communication between clients and software designers. Communication between clients and designers often "breaks down" because the two belong to different "work cultures;" in each of which they use their own vocabularies and symbol systems. We have developed EVIDII (an Environment for Visualizing Differences of Individual Impressions) as a cross-cultural communication medium for such collaboration. EVIDII visualizes relationships of three sets of data — persons, (visual) images, and affective words (such as "refreshing" and "warm"). EVIDII supports three types of interactive functions (*maps*, *perspectives* and *viewers*) allowing designers and clients (1) to become aware of differences of individual impressions of images and the use of words, and (2) to motivate them to actively discuss their differences. User studies have indicated that communication between clients and designers is enhanced through constructing shared communicative environments by using EVIDII.

KEYWORDS

Cross-cultural communication, Visualization, Human-computer-interaction, Software design

INTRODUCTION

Design tasks in complex domains are intrinsically collaborative. Complexity in design arises from the need to synthesize different perspectives on a problem. These perspectives originate from different work cultures, such as those of clients and designers. The challenge in cross-cultural design is to achieve shared understanding between groups of people that see the world in fundamentally different ways. System development is difficult "not because of the complexity of technical problems, but because of the social interaction when users and system developers learn to create, develop and express their ideas and visions " [5].

Two areas of research have developed to address the issues of collaboration and communication in design. *Participatory design* approaches try to give clients a voice in design [2]. Most do so by either requiring the clients to operate in the designers' world or the designers to operate in the clients' world. Both of these alternatives underutilize the skills of either clients or designers. *Computer-supported cooperative work (CSCW)* approaches often emphasize the use of computer tools to support physical coordination among stakeholders from a common culture working on shared materials, possibly at the same time [6].

This paper presents our approach to supporting collaborative design among stakeholders from different work cultures. Rather than forcing stakeholders to operate outside of their own cultures, we support them to construct a shared communicative environment [7] that bridge between the cultures. Rather than emphasizing physical coordination among designers, our approach emphasizes conceptual coordination among designers and clients. We define conceptual coordination as a process (rather than a state) of establishing and maintaining a shared understanding among the stakeholders in a design project.

EVIDII [9] (an Environment for Visualizing Differences of Individual Impressions) supports effective communication processes through creating shared understanding by visualizing differences among individual impressions of images and words. Using EVIDII, each client and designer

associate (visual) images (such as photographs and graphic images) with affective words (such as “refreshing” and “warm”). The system then visualizes relationships among the three sets of data — persons, images, and words — in a two- or three-dimensional space. By interacting with the visualization interfaces, both clients and designers gradually develop a shared understanding by asking questions such as “what does the client mean by using the word *pretty*,” “how does the client think of this particular image,” or “which designers find this image *cool*.” EVIDII provides a shared communicative environment, where the stakeholders can ground their communications for software design [7].

In this paper, we first discuss issues and challenges in supporting communication in software design. We argue for the importance of developing shared understanding and mutual knowledge through client-designer communication in design activities. We then present our approach and introduce the EVIDII system. Two scenarios illustrate how the EVIDII system supports the development of shared understanding and knowledge between clients and software designers by visualizing differences in impressions. The paper concludes with discussions of the approach.

IMPORTANCE OF COMMUNICATION IN DESIGN ACTIVITY

Design is collaborative in nature [4]. The predominant activity in designing complex systems is the participants teaching and instructing each other [5]. Because complex problems require more knowledge than any single person possesses, communication and collaboration among all involved stakeholders are necessary. Clients understand a problem and designers know how to solve the problem. Rittel termed this situation “symmetry of ignorance” [10]. That is none of these carriers of knowledge can guarantee that their knowledge is superior or more complete compared to other people’s knowledge. To overcome the symmetry of ignorance, as much knowledge from as many stakeholders as possible should be activated with the goal of achieving mutual education and shared understanding.

This communication poses two challenges: (a) neither clients nor designers can completely articulate what they want and what they know, and (b) communication between designers and clients sometimes breaks down because they use different “languages” [3].

Clients and designers belong to different “work cultures” [1]. Bodker and Pedersen [1] point out that an organizational culture can be observed through physical “artifacts” (such as office layout, decoration, work tools and dress code), “symbols” (such as stories, sayings, jargons, anecdotes and metaphors), and “work practices” (such as work routines, mode of cooperation, gestures and rituals). Cultural manifestations are easy to obtain but difficult to interpret, because they are ambiguous and may hold multiple meanings and understandings. Clients and designers have developed their own value systems and beliefs within their own cultures. Meanings of words may differ between cultures [11], and those who are outside of a certain culture may not necessarily understand a representation. When people who are collaborating do not share the same culture, knowledge, values, and assumptions, mutual understanding can be especially difficult. Such understanding is possible only through “the negotiation of meaning” [8].

In software design, clients and designers need to perform cross-cultural collaboration and, therefore, need such an environment as gradually constructing mutual and shared understanding.

The goal of our research is to support this cross-cultural communication process between clients and software designers. Instead of trying to develop a stable ontological mapping between two languages, our approach is to use a computational environment that makes them aware of the existence of differences in their expressions for representing impressions. Once they become aware of the differences, people are good at using the breakdown as an opportunity to develop further shared understanding.

The next section describes EVIDII (an Environment for Visualizing Differences of Individual Impressions), which supports cross-cultural communication by helping clients and software designers in constructing shared understandings through finding and recognizing differences of “impressions.”

THE EVIDII SYSTEM

EVIDII first asks users to associate (visual) images with affective words. Then, the system provides interactive interfaces that visualize the relationships among the three sets of data — persons, images, and affective words.

Communication between clients and software designers using EVIDII proceeds as follows: Both clients and

software designers associate images with words using EVIDII. By using EVIDII, they may discover differences in the associations, such as the use of words or the use of images. This discovery motivates them to talk about the differences. By iterating the above two steps, they gradually develop shared understanding.

In this section, we first briefly describe the functionality of the EVIDII system. We then describe how users find differences in their impressions using EVIDII.

Functions of EVIDII

The EVIDII system deals with two sets of "objects (e.g., images and words)" and a set of "people." EVIDII allows users to survey how "people" think about a set of "objects" or how "people" associate objects from one set of objects to another, for example, how people think of images. In this paper, we use a set of visual images and a set of affective words as the two sets of objects.

The *Word List Editor* (Figure 1-(a)) allows users to specify the set of words that are to be used (listed in the left side of the Word List Editor). Users can add, modify or remove words from the set of words. The *Image Object List Editor* (Figure 1-(b)) allows users to specify the set of visual images in the GIF or JPEG format. Figure 1-(c) shows the *User List Editor*, with which users can be registered. Each user is assigned an icon image, which is used when visualizing the relationship between the objects. Figure 1-(d) is the *User Profile Editor*, with which each user can associate images with more than one word.

The EVIDII system provides three types of interactive functions:

- *maps*,
- *perspectives*, and
- *viewers*.

These functions are used to visualize the two sets of objects and the set of people, along with their relationships. Users can "discover" new relationships as well as examine a specific relationship in more detail from what they have discovered.

Maps

Each *map* is a visualization of the set of objects in a two- or three-dimensional space and provides a basis for how users can view the relationships among the sets of objects. A map can be either subjective or objective. An objective map is based on computationally derivable properties of the set of objects. Taking a set of images for instance, an example objective map would use the HSB (Hue, Saturation and Brightness) value of the most frequently used color of each image as the three-dimensional coordinates. Each image, then, would be positioned on the map according to the coordinates. A subjective map has the users decide on where each of the objects should be positioned within the two- or three-dimensional space. Taking a set of affective

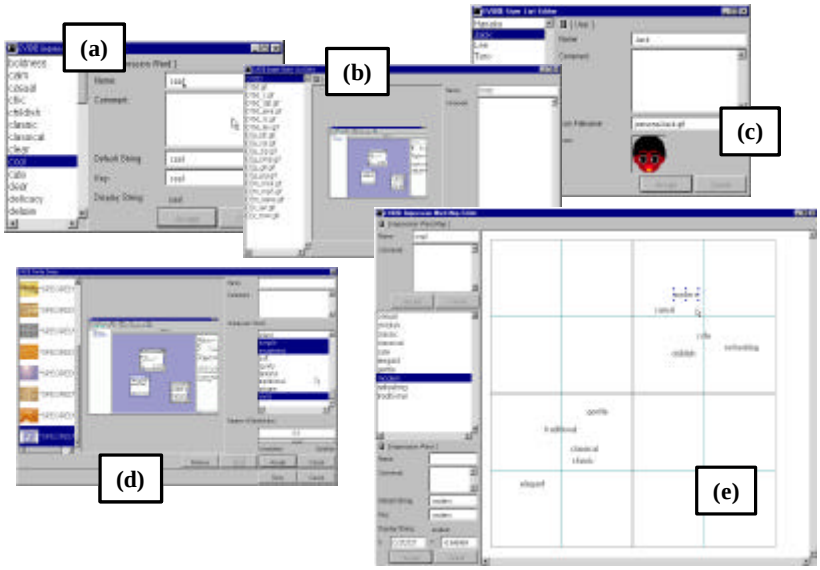

Figure 1: (a) Word list editor (b) Image object list editor (c) User list editor
(d) User profile editor (e) Subjective map editor

words for instance, “cool” and “cold” can be positioned far away from “warm” and “hot” in a two-dimensional space. Figure 1-(e) shows the *Subjective Map Editor*. Users can position the set (or subset) of the words, that were specified in the Word List Editor, in the two-dimensional space in the direct manipulation style.

Perspectives

Users can change the perspective or how they look at the data to understand the relationships among the sets of objects. For example, if the user initially takes an *image perspective*, the user can find “the person who selected this image” and “the words that were associated with this image.” The user can then change the perspective to *person*, in which case the user can examine “the words that were associated with this image by that person” in more detail. The user can also change the perspective to *image*, in which case the user can examine “the persons that associated this word to this image.”

In this way, the perspective function allows users to change how they look at the data according to what the users want to know. Users can understand not only the relationships among the three sets of data but also characteristics of the sets of objects as a whole. They can further understand more minute characteristics concerning specific objects.

Viewers

Viewers are used to display a visualization of the relationship among the sets of objects on a particular map. Each viewer allows users to take a certain map and certain perspectives. When the user changes perspectives, the viewer dynamically changes the visualization. Figures 2-(a), (b) and (c) show viewers taking various perspectives and maps. The top-left window of a viewer shows a list of maps. Users can select a map by clicking on one of the maps. The middle-left window lists a set of images and the bottom left window shows a list of names representing persons. Clicking on one of the images or on one of the individual list allows users to select an *image perspective* or a *person perspective*. Figure 2-(a) shows an example of a viewer that shows the results when taking an *image perspective* by selecting one of the images, while Figure 2-(b) shows a viewer using the same map taking a *person perspective* by selecting one of the persons. Figure 2-(c) meanwhile shows a viewer taking a *person perspective* but using a different map.

USING EVIDII IN SOFTWARE DESIGN PROJECTS

This section presents scenario illustrating how EVIDII can be used to support a variety of software design processes, and results of our user observations.

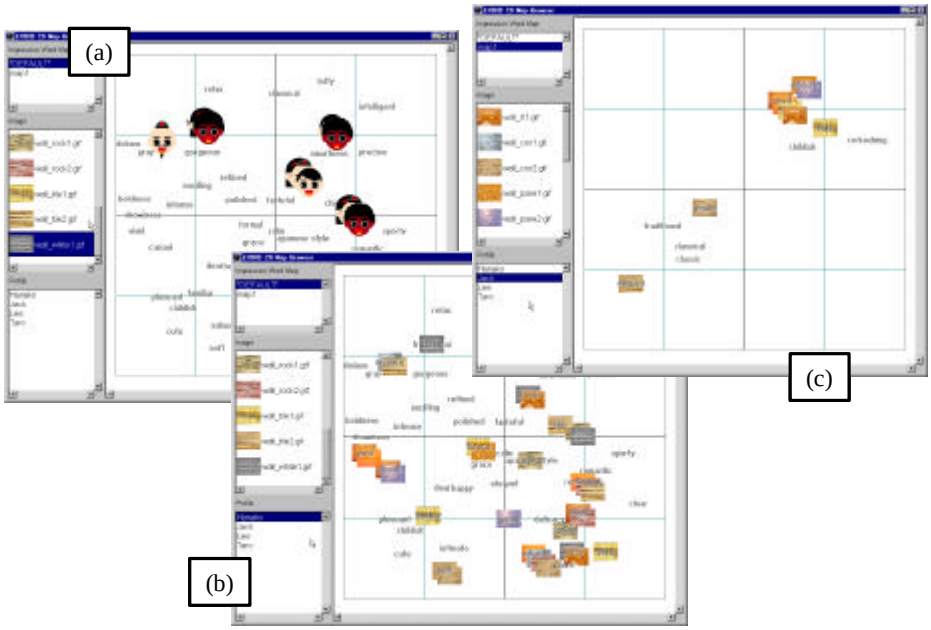

Figure 2: (a) Viewer taking an image perspective (b) Viewer taking a person perspective
(c) Viewer using a different map

Two scenarios

Let us take the graphical user interface (GUI) design, as a typical collaborative situation between clients and designers demonstrating how EVIDII supports their cross-cultural communication.

1. Clients and/or designers (henceforth users) first use the Image Object List Editor to input a set of sample GUI screenshots of a variety of software (e.g., word processors, spreadsheet programs, mailers, etc) as visual images.
2. They then use the Word List Editor to specify a set of words that are concerned with quality of software, such as “useful,” “efficient,” or “functional,” as well as those concerned with ambience of how they look, such as “cool,” “simple,” or “cute.” They then register themselves in the system using the User List Editor.
3. With the User Profile Editor, each user associates words with images, i.e. chooses words that is deemed to “appropriately” describe each screenshots image that EVIDII displays.
4. The users choose a map and visualize the results on a viewer.
5. The visualization may cause the users to see differences between how each of them “feel” about certain screenshots. This will then cause them to ask questions (such as “why did you have such an impression with the look of the screenshot?”) to understand each other more deeply.

In this case, suppose the client had a vague requirement, i.e. “the GUI should be *cool-looking*.” The designer needed to understand what the client meant by “cool.” What sort of feeling did the client have in mind when using the word “cool?” They alleviated this problem by using the *perspective* function in EVIDII. The client and the designer examined the screenshot images that each had thought to be “cool.” This resulted in their understanding what types of images each of them considered as “cool.” They further took one of the images and discussed what caused the client to consider that to be “cool” in more detail.

As another case, suppose a client has a certain set of favorite GUI components (e.g., icons, layouts, menu appearances) from existing software applications. Since the client does not necessarily have GUI design knowledge about how to combine those components, however, simply using those favorite ones may possibly result in a chaotic design that lacks coherence. This problematic situation can be solved by using EVIDII.

The client first sets as visual images in EVIDII a variety of GUI components including their favorite ones. Then, each of the client and software designers associates the images with the words set up in EVIDII a priori.

After the survey, the client and the designers use EVIDII to examine words that had been associated with icons and layouts that are the client’s original favorites. They would then be able to understand that certain combinations of GUI components would not lead to the effect that the client had originally intended.

User observations

We observed that both clients and designers using EVIDII could identify differences in how they used and felt about words and images, and then were motivated to “talk about” the differences resulting in a shared understanding about their design tasks. By using three types of interactive functions of EVIDII (*maps*, *perspectives* and *viewers*), users could closely look at “what surprised” them, and verbally ask follow-up questions to each other. This then resulted in asking further questions and searching for other interesting relationships. This process was repeatedly observed during the user study sessions.

In short, EVIDII evoked users to become aware of word-image associations that “surprised” them, which can then be further examined through various types of viewers integrated within EVIDII. This guides the design stakeholders in conducting smoother communication and helps them to develop shared understanding.

DISCUSSION

In software design, software designers need to keep focusing on the following two questions:

- What do clients want?
- How do clients like the produced artifacts?

To address the first question, clients and designers use explicit representations (such as mockups and prototypes) to communicate their intent. However, this approach sometimes does not work because there can be “different images” of such external representations by each stakeholder.

To illustrate this point, an interesting anecdotal story was told by one of the software designers we have interviewed. In one design meeting, a concept of design was agreed

among designers that "our product would be like Mickey Mouse." While Mickey seemed to be a concrete enough representation to communicate the intention, one designer thought he had to use bold lines "like Mickey" and another designer thought that she had to use round forms "like Mickey." The representation of Mickey has many aspects: shape, the thickness of lines, colors, to name a few. When the designers agreed on the use of Mickey for their concept design, each designer was focusing on different aspects of Mickey resulting in a communication breakdown.

By using EVIDII in ways illustrated in previous section, it is possible to explore clients' requirements in a more concrete manner; for example, "what functions do clients need or not need?" and "what 'taste' of software do clients want, Windows-like, Unix-like or Mac-like?" before actually starting design.

EVIDII can also be used to address the second question. This issue arises when clients tend not to dare to articulate what they think of proposed artifacts by professional designers because they feel "incompetent" to do so. There is an inevitable prejudice that "nonprofessional clients should not argue against professional designers' decisions." This often results in the lack of immediate feedback from clients about intermediate designed artifacts, which would have played a critical role for software designers in carrying out subsequent design processes.

EVIDII serves as a communicative environment that addresses this issue. There are no arguments concerning "better" or worse around the associations made by the design stakeholders. Instead, EVIDII reveals the existence of differences of impressions among the stakeholders. This is a type of feedback that designers can obtain from the client through EVIDII. This will inform designers how to proceed the rest of the design task by reflecting the client's intention more "correctly."

CONCLUSION

This paper presented EVIDII, an environment that supports cross-cultural communication in a variety of design situations including software design. We discussed the importance of communication in design activities and its difficulty based on differences in "culture." We introduced the EVIDII system as an environment that helps overcome this difficulty by focusing on the differences themselves. Taking GUI design as an example, we elaborated this point by showing example use situations. Our future work includes the application of EVIDII for more design situations as well as for other "cross-cultural" collaborative situations, the analysis of those use situations, and the refinement of the system based on those results.

ACKNOWLEDGMENT

We thank Shingo Takada at NAIST for valuable discussions.

REFERENCE

1. Bodker, K., & Pedersen, J. (1991). Workplace Cultures: Looking at Artifacts, Symbols and Practices. In J. Greenbaum & M. Kyng (Eds.), *Design at Work: Cooperative Design of Computer Systems*, 121-136. Hillsdale, NJ: Lawrence Erlbaum Associates.
2. CMCM (1993) communications of the ACM, Special issue: Participatory Design, June 1993.
3. Ehn, P. (1989). *Work-Oriented Design of Computer Artifacts. (Second ed.).* Stockholm: Arbetslivscentrum.
4. Fischer, G., Nakakoji, K., & Ostwald, J. (1995). Supporting the Evolution of Design Artifacts with Representations of Context and Intent, *Proceedings of Symposium on Designing Interactive Systems (DIS'95)*, 7-15. New York: ACM.
5. Greenbaum, J., & Kyng, M. (Eds.). (1991). *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
6. Grudin, J. (1994). J.Grudin, *CSCW: History and Focus*, IEEE computer, Vol. 27, No. 5, 1994, pp. -27
7. Krauss, R.M., & Fussel, S.R. (1991). Constructing Shared Communicative Environments. In Resnick, L. B., Levine, J. M., & Teasley, S. D. (Eds.). *Perspectives on Socially Shared Cognition*, 172-200. Washington, D.C.: American Psychological Association.
8. Lakoff & Johnson (1980). *Metaphors We Live By*. Chicago and London, The University of Chicago Press.
9. Nakakoji, K., Yamamoto, Y., Sugiyama, K., & Takada, S. (1998). Finding the "Right" Image: Visualizing Relationships among Persons, Images, and Impressions, *Proceedings of Designing Effective and Usable Multimedia Systems (DEUMS)*, 91-102.
10. Rittel, H. (1984). Second-Generation Design Methods. In N. Cross (Ed.), *Developments in Design Methodology* 317-327. New York: John Wiley & Sons.
11. Robinson, M., & Bannon, L. (1991). Questioning Representations, *Second European Conference on Computer-Supported Cooperative Work.* 219-233, Kluwer Academic Publishers.

Appeared in Proceedings of the International Symposium on Future Software Technology (ISFST-99), Nanjing, China, Software Engineers Associates, pp.113-118, October, 1999.