UNCONVENTIONAL INSPIRATIONS
FOR CREATING
SOFTWARE INTERFACE METAPHORS

DINESH S. KATRE
Group Coordinator
Center for Development of Advanced Computing (C-DAC)
A Scientific Society under Ministry of Information Technology, Government of India,
Agriculture College Campus, Near District Industries Centre, Shivaji Nagar, Pune 411005, India.
Tel/Fax: 91+020+5533250 e-mail: dinesh@cdac.ernet.in

Abstract:

In order to come out of the typical mould of interface metaphors used for software, new inspirations are sought from *Dnyaneshwari*. *Dnyaneshwari*, one of the best Indian saint literatures uses numerous metaphors for depicting complex and abstract philosophical information. It is proposed to have variety of Interface Agents having different behavioral styles for guiding the software users. The choice of Interface Agent should change the interface metaphor for addressing varied user groups. Composite metaphors should build the dictionary and the metaphoric language. 'Metaphor-sensitive-software' that switches appropriate interface metaphor by observing the user traits and 'user-sensitive-metaphor' that perfectly represents the mental model of user are proposed. Shortcomings of one type of interface for all types of users are highlighted. Proper selection and gradation of interface rendering types such as- Life-like, Moderate, Subdued and No Metaphor are suggested. The importance of studying the metaphors used in conventional art forms is emphasized for creation of better software interface designs.

Key Words:

Dnyaneshwari, Software Interface Metaphor, Interface Agents, Mental Model of User, Cross-Domain Mappings, Desktop, Types of Interface Rendering, Composite Metaphors

Various Forms of Metaphoric Expressions:

We have always used metaphoric expressions in all forms of communication for presenting the message effectively. Various types of literature such as- poetry, humorous and satirical writing, spiritual and religious writing and even the day-to-day conversations make use of metaphoric expressions. Indian classical music uses metaphors, which help the musician in the rendition of different emotions, moods and situations. The theatrical plays, the dance forms e.g., *Bharatnatyam* and the architectural designs make use of spatial metaphors. Paintings, sculptures, murals and even the movies have used metaphors for depiction of abstract concepts. Haptic metaphors are now being explored especially for visually impaired persons. Basically, the point established here is that metaphors are integral part of any communication and we use them knowingly or unknowingly irrespective of our academic backgrounds and literacy levels [8]. This happens primarily because of the thinking and comparing nature of human beings.

Application of Metaphor in Software Interface Design:

Real life metaphors are used for designing the software interface. Basically, the software interface grows around the metaphor. It helps in evoking the examples from real life and matching the software with the mental model of user [4]. The interface metaphor gradually takes the user from known to unknown aspects of software. Metaphor is essentially a comparison between reference and application domains [3, 7]. It highlights the similar qualities of application with help of the reference domain e.g., the 'Trashcan' from the Windows Desktop quickly enables the user in understanding the concept of deletion of files. The merit of interface metaphor depends on the number of cross-domain connections (Similar aspects between reference and applications domains) [1].

Unconventional Inspirations:

My closest encounter with Metaphors happened while working on the Multimedia Rendering of *Dnyaneshwari* [1] project. There is much written about application of metaphors in software designing. Having gone through most of the research work in this area, it is felt that the metaphors from *Dnyaneshwari* have a lot to offer for software interface designers. Lets go through some of the metaphors from *Dnyaneshwari.* After describing the metaphors, their certain qualities applicable in software domain are elaborated.

<u>Metaphor 1.</u>

Saint *Dnyaneshwar* has used various metaphoric identities such as- Blind, Handicap and Mad persons. Whenever he talks about the blind person, it actually means an ignorant person, one who has eyesight but is unable to notice the necessary things. It is actually not related with physical blindness. He addresses an inactive person as handicap, one who knows the problem but prefers not to act. It has nothing to do with physical disability. One, who is completely driven away by his greed, is addressed as a mad person. In the philosophical discussions, he uses them as metaphoric representations. If you interpret them literally then the meaning would get distorted.

Interface Agents:

Above example from *Dnyaneshwari* is very similar to 'Interface Agents' used in software. Interface agent is basically the personified character of user interface. The computers behave and its application software can have predispositions to behave in certain ways on both functional and stylistic levels [9]. The three characters mentioned earlier are very symbolic and keep appearing very consistently throughout the Chapters of *Dnyaneshwari*. They highlight different details of their personality depending on the context. It is possible to visualize interface agents such as guide, helper, teacher, watchman, doctor, troubleshooter, etc., that can influence the software.

[1] *Dnyaneshwari* is one of the most profound commentaries on *Bhagavad-Gita*, a Hindu religious/spiritual scripture. It was written by Saint *Dnyaneshwar* around 700 years ago.

Some interface agents have already been designed and introduced in software e.g., MS Word has an ‘Office Assistant’ that helps the user. One can choose a character of his/her choice such as Albert Einstein or some abstract characters and animal caricatures as ‘Office Assistant’. But they lack the behavioral details and personality. They also do not have their unique styles of helping. These agents are more like animated pictures that link you to standard software help.

There could be interface agents with different personalities and roles defined for them. Their unique behavioral aspects could help in satisfying different age groups, cultural and academic backgrounds and ego states of users. Presently, change in the choice of interface agent does not change the interface metaphor. **It will be ideal to change the interface metaphor depending on the choice of interface agent e.g., if one chooses ‘Teacher’ as the interface agent, it conveys that the user is a learner and as a result the interface metaphor should also change to suit the learner’s requirement.** It should create more suitable environment for learning.

Metaphor 2.

Saint *Dnyaneshwar* has referred sunrise as the grace of *Guru* (Spiritual Master) or enlightenment and the sun as a representation of *Guru*. The lotus is a representation of human body and the beetle, which is in love with the lotus, is like the soul. The night represents ignorance of a person and the twinkling stars are like the material knowledge. In the night, the lotus closes all its petals and the beetle gets locked inside. Love or affection is very tender to feel but very to hard break. He says, the beetle can penetrate through a hard wooden log but prefers to be caught and suffocated inside the closed petals of lotus. The soul represented as beetle is freed when the sun rises and the lotus blooms. The sunrise takes away the night of ignorance as well as the twinkling stars of material knowledge.

Metaphor 3.

The river represents the subject of knowledge and the banks of river are like the division of opinions and viewpoints. There is night of ignorance. There is a pair of birds, which is separated on the banks of river. It is unable to see each other due to darkness. The pair of birds represents the spiritual knowledge and the devotee who is in search of it. They both are able meet when the sun rises. It is already mentioned above that the sun represents *Guru*. With the sunrise or by the grace of *Guru*, the spiritual knowledge and the devotee meet each other and the devotee gets enlightened.

Dynamic Qualities of Metaphors:

There are hundreds of such metaphors, which are interwoven in the philosophical discussions of *Dnyaneshwari*. Saint *Dnyaneshwar* has used them very consistently throughout his writing. The beauty of all these metaphors is that they are perfectly networked for building complexes of information. While reading *Dnyaneshwari*, for a beginner, it is very interesting to get introduced to the metaphoric representations. At a later stage of his reading, he is able to interpret the apparently picturesque but very abstract in the core and philosophical information. Entire philosophical interpretation of ‘sunrise’ comes to your mind the moment ‘sunrise’ is mentioned. These interpretations are very long but compressed in just few words.

A lyrical verse written by Saint *Dnyaneshwar* consisting about 10 to 15 words can be interpreted in several pages of text. The experienced and knowledgeable philosophers who understand Indian Spiritualism can do this. The metaphors narrated above create very picturesque, very apt and very involving imageries in the minds of readers. The prismatic effect is felt when two or more metaphors are brought together in a situation. Metaphors are used in different situations and in different combinations to highlight new meanings and transmit messages. The imaginative and colorful application of metaphor is contributing its own color to the meaning and making it more interesting. Saint *Dnyaneshwar* has exploited the dynamic qualities of metaphors throughout his writing.

Metaphoric Language:

Primary objective of Saint *Dnyaneshwar* in using metaphoric language was to take the spiritual knowledge to laymen and present it in the most simplified manner. The metaphors serve as interfaces to abstract and complex philosophical information. Virtually, he has built a ‘large dictionary’ of such metaphors through his writing. You see this dictionary eventually constructing the metaphoric language. The objective of using interface metaphors in software design is also no different. The basic metaphors should be such that they can be interlinked, networked and structured. This makes all the basic metaphors as the dictionary of metaphoric language (in specific domain).

Building Scenarios using Multiple Metaphors:

Metaphor 2. & 3. are basically a composition of various metaphors that build a scenario or a situation. The metaphors carry their individual symbolism and at the same time they communicate much wider message in togetherness. The software metaphors lack this richness. **Application of metaphor in software is very static, as when more metaphors are brought together the scenarios do not get built** e.g., the desktop remains a desktop irrespective of whether one is using it for video editing or for 3D modeling or for textile designing. The three activities given here are not performed in office like environment but they continue to get represented as desktop activities.

Desktop Hangover:

In the 1970s, the modern graphical user interface was invented at Xerox Palo Alto Research Center (PARC). Apple Macintosh successfully implemented this GUI with its Desktop metaphor [2]. The desktop metaphor was found most suitable because the computer was mainly doing the office work. It was dealing with documents, files and folders. The same metaphor continues in Macintosh or Microsoft Windows operating systems even today.

The computer today, plays the roles of Video Production House, Design Studio, Animation Studio, Industrial Workshop, Training Centre, Museum Archival, Doctor’s Clinic, Stock Exchange, Post Office and so on. The present form of Windows File Manager recognizes diverse file types such as video, audio, text, images, 3D models, and so on, but maintains them as documents on the desktop. It shows that the desktop metaphor has not evolved further and the hangover continues. In real life, we have bookshelf for keeping books, racks for keeping audio and video tapes, gallery for displaying pictures, fridge for keeping eatables, safety-vaults for keeping jewelry, etc. The system does deal with wide variety of contents for diverse types of activities but continues to represent them as files and folders on the desktop. May be the File Manager should change to 'Content Manager' and represent various content types in unique manner and not just as files and folders.

Metaphor-Sensitive-Software & User-Sensitive-Metaphor:

Presently, the software offers single and common user interface for all types of users. In this approach, it is not possible to have true user orientation. The software developer may be able to satisfy only one kind of user or none of them. It is high time to have rich and dynamic software interfaces. The metaphor-sensitive-software is the need of today and it will be definitely achievable with the powerful Windows XP 64-Bit Edition. The 64-Bit applications can easily accommodate far better visual quality, animations, multimedia interfaces and dynamic metaphors. Following are the qualities of proposed metaphor-sensitive-software-

i. It should have interface agents addressing to various age groups and ego states, cultural and academic backgrounds of users.

ii. Choice of interface agents should change the interface metaphor and certain behavioral aspects suiting to the type of user.

iii. The interface metaphor should provide necessary variations to address different user levels such as novice, proficient and expert. Also, the interface for male and female users could be different.

iv. The users should be able choose the type of interface metaphor depending upon their need when they purchase the software. They should be able to plug a new interface metaphor in the already existing software.

The 'metaphor-sensitive-software' is expected to give appropriate metaphoric response to user depending on the type of user. The 'user-sensitive-metaphor' should change the user interface for matching it with user's mental model. This will help in building scenarios out of interface metaphors.

Following metaphor is presented by giving interactive links to relevant information.

<u>Metaphor 4.</u>

In the final, eighteenth chapter of *Dnyaneshwari*, Saint *Dnyaneshwar* says that this chapter is like the pinnacle of the *Bhagvad-Gita* Temple. From this highest point you get the Arial view of previous seventeen chapters. Then he describes earlier seventeen chapters by linking them with the remaining architectural parts of temple. Other philosophical literatures such as *Upnishadas, Vedas, Mahabharata, Navrasas,* etc., are visualized while describing the premise and surroundings of *Bhagvad-Gita* Temple.

Following advantages are observed in this metaphor for interaction design:

- It is possible to provide links all over the illustration with logical connectivity.
- Extremely appealing, eventful and easy to recall.

- Interactivity through metaphor helps in clearly presenting the reference and application domains of metaphor. It does not leave much scope for ambiguity unlike purely linguistic metaphors.
- It mobilizes the users imagination.
- It makes user intuitive and prompts him in further navigation.
- It captures and depicts more information.
- Reduces the cognitive load on the users.
- Provides a conceptual model for mapping the information.

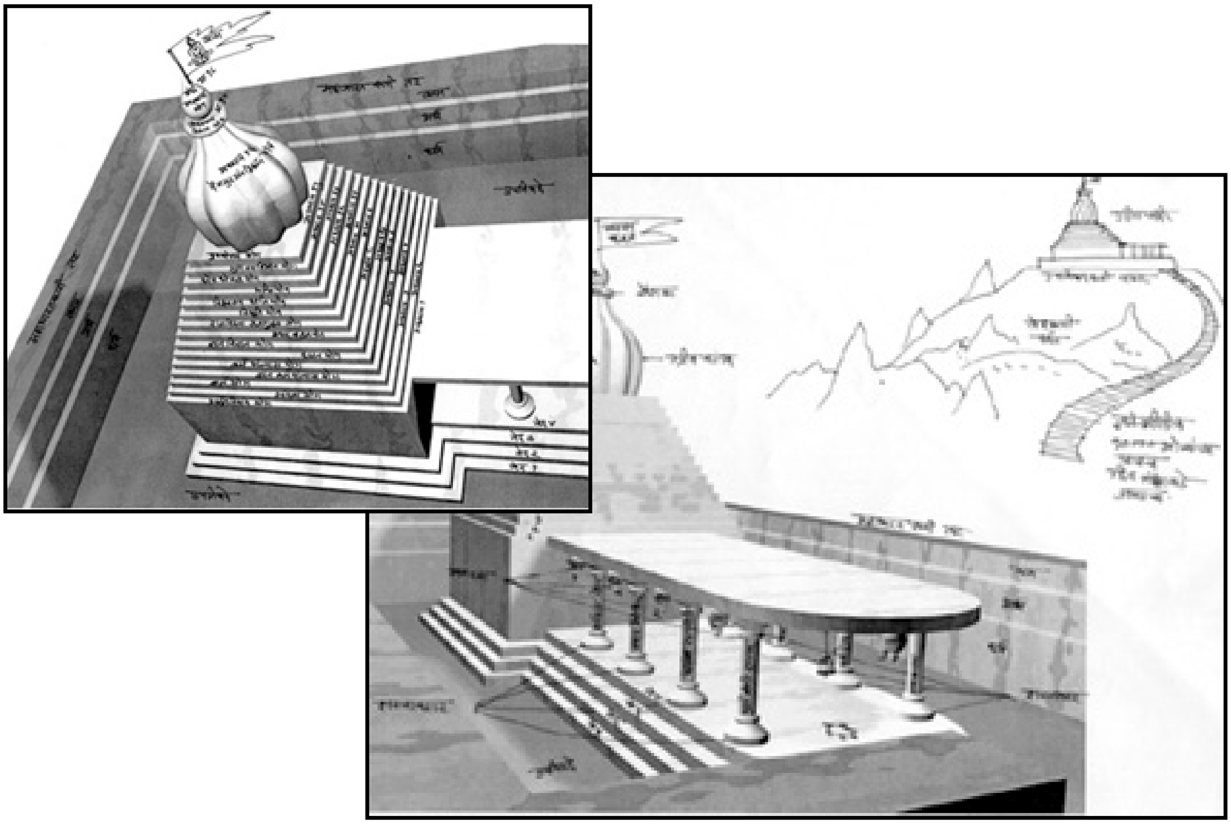

Diagram 1. Linking structure in the form of Bhagavad-Gita Temple

Above-mentioned metaphor is applied only at the surface level of information. It is providing a conceptual model for presenting an index for various chapters of *Dnyaneshwari* and other sources of information. There are no iconic representations but the entire structure of temple is mapped with links. Such application of metaphor may be useful for designing the home pages of websites. This metaphor is quite different than the usual interface metaphors. The temple metaphor provides a conceptual layout for providing various links but it does not extend further in the sublevels of information. A 3D model of temple is generated for giving links in three-dimensional environment. We propose to use VRML authoring for this purpose.

Comparison of metaphors from *Dnyaneshwari* and Software Interface Design:

One may feel that there is greater flexibility for using metaphors in a philosophical subject. Software has to convey and deliver very specific output. But there is tremendous scope for improvement in the application of interface metaphors. The application of interface metaphor is not as bold as the linguistic examples described above. Most metaphors used in software design are very predictable and realistic. They lack the element of surprise and innovative connections to different ideas. This could be because the software interface designers are still in the elementary stage. The interface metaphors explore the look-alike aspects more and their application is very subdued, conservative type. This causes less amount of abstraction of information.

Antonymic metaphors are not used in software e.g., “if the night is ignorance, the sunrise becomes enlightenment”. The second half of the statement can be inferred and does not require specific mention of it. The higher-level metaphors can offer innovative concepts and greater abstraction of information. Increased participation of interface designers in software designing process can improve this situation.

Metaphors of *Dnyaneshwari*, in our prevailing terminology, can be called as composite metaphors. Basically, multiple metaphors are used for representing the structure of information. Most interface metaphors communicate one message per connection in the metaphor. They do not emit new meanings and messages in different situations. Basically, the software does not change the metaphor or metaphoric response depending on the situation.

Manifestations of Metaphoric Interfaces in Software:

In order to bring dynamic changes in the metaphoric response of software, it is necessary to know various types of rendering and manifesting the interface. One should be able to select suitable approach for its detailing and presentation. The rendering of interface metaphors can be classified in terms of four categories such as- Subdued, Moderate, and Life-like and No Metaphor. Appropriate approach to rendering of interface can make the technology transparent and also hide the details of implementation [6].

Subdued / Conservative Interface:

The popular Desktop metaphor can be called as Subdued / Conservative Interface metaphor. If you ask a novice to browse through Macintosh or Windows OS, he may not immediately realize its conceptual familiarity with ‘Desktop’. But if he is introduced to the ‘Desktop’ concept then he might feel the resemblance between the real desktop and the interface of the OS. This is because the application of desktop is at the concept level and some amount of physical form is given to express it. The appearance of desktop is not obvious.

Advantages:

- Even if the metaphor remains unchanged in a particular situation, it does not stand out very conspicuously.
- The outgrown portions of software that are difficult to cover in the metaphor do not standout as misfit portions of interface.
- It is easy to adopt composite metaphors as the contradiction or clash between two metaphors can be handled in a subtle manner.

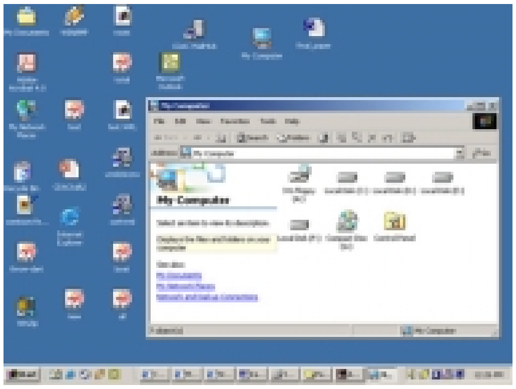

Diagram 2. Subdued interface of Windows Desktop

- More suitable for experienced users as the subdued rendering of interface stresses more on the conceptual matching with software.
- This type of interface can be used in both static and dynamic manner, as it does not have very fleshy physical form.
- The software may not require very highly configured system.
- Suitable for complex and procedural software.

Disadvantages:

- It may not necessarily simulate the desired user environment.
- It may not offer a distinct identity to software.
- Not so good for beginners.

Moderate Interface:

Most of the MP3 Players or the Digital Camera software carry moderate interfaces. The control panels of MP3 Players very much resemble the physical audio systems. The Digital Camera software looks like a photo studio. The moderate interfaces are depicted in the most realistic manner. But the proportion of objects is manipulated depending on the layout compulsions. They are suitable for software that has very focused and limited features to offer. Most multimedia applications carry moderate manifestation of interfaces. The metaphors rendered in *Dnyaneshwari* are also quite moderate in terms their presentation [5].

Advantages:

- Extremely inviting and best suited for novice computer users.
- Believable simulation of user environment.
- Suitable for software, which does not involve long and complex procedures.
- Good for software where the user is not expected to learn but to fiddle with the controls and figure out how the software works.
- It does not require highly configured system.

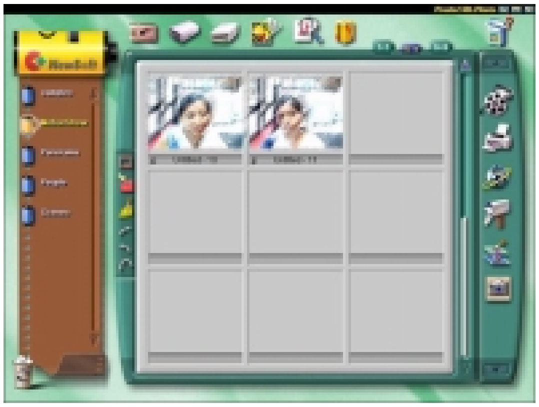

Diagram 3. Moderate interface of photo capturing software

Disadvantages:

- The moderate metaphors are quite difficult to merge or combine with other metaphors as they have very fleshy physical form.
- Not suited for complex software offering diverse features and utilities
- This type of interface cannot be very dynamic due its articulated physical form.

Life-like Interface:

Interfaces are vividly represented in very specialized application domains such as computer games, simulators and virtual reality applications. One may see highly realistic objects and environments used as interface for user interaction. Special types of sensing devices are used for interacting with software.

Advantages:

- Offers impressive effects and experiential navigation.
- Believable simulation of user environment.
- Similar to real world interaction.
- Good for highly specialized and entertainment applications.

Disadvantages:

- Suitable for very specialized and limited applications.
- Not suited for complex software offering diverse features and utilities.
- Requires very advanced and expensive equipments.

Most software tools manifest either subdued or moderate interfaces. What is preferred is the right combination of moderate and subdued rendering types. Software should have properly controlled gradation of Life-like >> Moderate >> Subdued >> No Metaphor rendering. 'No metaphor' must be considered as it is not possible represent all the procedures of software in a metaphor. Life-like / immersive interfaces are very exceptional in nature and therefore they are not required for regular software.

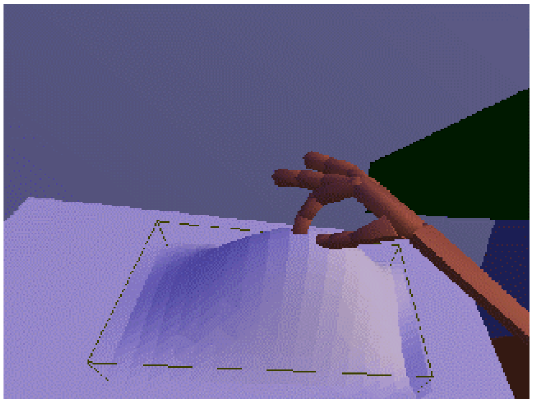

Diagram 4. Life-like interface of VR Application

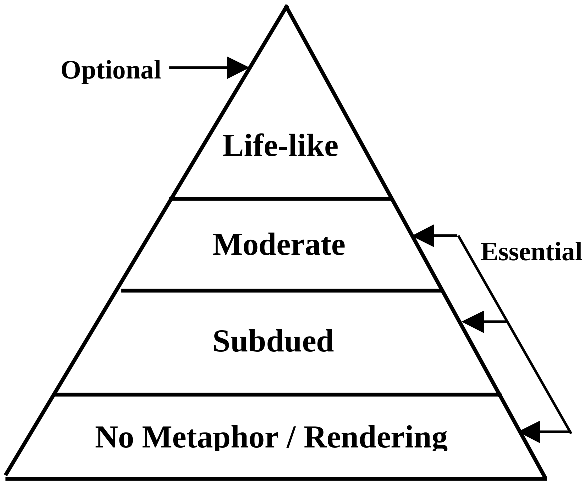

Diagram 5. Gradation of Interface Rendering Types

Conclusions:

i. The metaphoric expressions in the prevailing art forms such as- literature, paintings, music, etc., should be studied for their appropriate application in software interface design. It will help in humanizing the software interface further.

ii. Choice of 'interface agents' should change the 'interface metaphor' and certain behavioral characteristics suiting to the type of user. The user should have choice of selecting the interface agent from the library provided along with the software.

iii. It is necessary to develop 'Metaphor-sensitive-software' that switches appropriate interface metaphor by observing the user traits and 'user-sensitive-metaphor' that perfectly represents the mental model of user.

iv. The interface metaphors should be rendered with proper selection and gradation of rendering types such as- Life-like >> Moderate >> Subdued >> No Metaphor so as to accommodated the desired dynamism of metaphors.

References:

Journals / Proceedings

[1] Selections from "The Contemporary Theory of Metaphor" by George Lakoff in Ortony, Andrew (ed.) Metaphor and Thought (2nd edition), Cambridge University Press, 1992.

[2] Alan Cooper, “The Myth of Metaphor”, Published in Visual Basic Programmer's Journal, USA, July 1995.

[3] Ruth A. Palmquist, “The Search for an Internet Metaphor: A Comparison of Literatures”, ASIS Conference Proceedings, Oct. 19-24, 1996.

[4] Ken Mohnkern, “Visual Interaction Design: Beyond the Interface Metaphor”, The SIGCHI Bulletin, ACM Press, Vol.29 No.2, April 1997.

[5] Dinesh S. Katre, “Multimedia Rendering of Spiritual Texts” Proceedings of Virtual Systems and Multimedia ’99 International Conference Proceedings, pp. 500-507, University of Abertay, Dandee, Scotland, UK, Sept. 1999

[6] Frank M. Marchak, The Magic of Visual Interaction Design, The SIGCHI Bulletin, ACM Press, Volume 32, Number 2, April 2000.

[7] Dinesh S. Katre, “Template based Authoring for Cost Effective Multimedia Production”, IEEE Multimedia Technology and Applications International Conference Proceedings, pp. 133-142, University of California, Irvine, USA, Nov. 2001.

[8] Aaron Marcus, “Metaphors and User Interfaces in the 21st Century”, Interactions of ACM, Volume IX.2, March+April 2002.

Books

[9] Brenda Laurel, Interface Agents: Metaphors with Characters, The Art of Human-Computer Interface Design, Addison-Wesley Publishing Company, inc. USA. 1990.

Curtsey:

- Microsoft Windows 2000
- Presto! Photo Software
- Virtual Reality Applications Center (VRAC), Iowa State University, USA

Acknowledgements:

- R. K. Arora, Executive Director, C-DAC for permitting me to pursue higher studies and encouraging my research explorations.

- Dr. Mukul K. Sinha, my Ph.D. Supervisor and Managing Director of Expert Software Consultants, Pvt. Ltd. for his valuable guidance.