FROM SIMULATION TO STIMULATION

Stimulative Design Computing Learning & Teaching Environment based on Multiple Representations

ALPHA WAI KEUNG LEE
Key Centre of Design Computing and Cognition
Faculty of Architecture, University of Sydney
alpha@arch.usyd.edu.au

Abstract. Design computing is an interdisciplinary field that centers on the intersection of design, computer science, and cognitive science. These multi-dimensional faces of this new paradigm necessitate a mutual understanding of computing and design as a whole, which are usually conceived as separate stages of computer-mediated design. This paper explores the use of multiple representations in stimulative design computing learning and teaching environment. This paper starts with an overview of computer-mediated collaborative learning, multiple representations and learner-centered design. Then issues related to the methodology in implementing the system are discussed. Agent models and prototype of the above computer-mediated collaborative learning system are constructed. The research is under progress.

1. Computer-Mediated Collaborative Learning

In recent years, educational research had attempted to determine under what circumstances collaborative learning is more effective than learning alone, and more recently, numerous studies have focused on computer-mediated collaborative learning. In psychology, interest in collaborative learning is related to the emergence of new theories such as “share cognition” and “distributed cognition”. These theories move away from the view traditionally held in cognitive science according to which human cognition is bound inside individual heads. The word “collaboration” is also used very frequently in computer science to describe the interactions among artificial agents (Dillenbourg, 1998).

2. Multiple Representations

In learning declarative knowledge and problem solving skills, multiple representations such as verbal, graphical and mathematical representations, knowledge at different levels of abstraction (e.g. qualitative and quantitative, specific cases or general models) are involved (van Someren et al., 1998). Seel and Dörr (1994) have discussed the distinctive relationship of supplantation of mental images through graphics and the instructional effects on spatial visualization skills of adults. Boshuizen and Schiff (1998) have discussed the issues of problem solving with multiple representation by multiple and single agents. This coordination of multiple representations is a topic that has not received as much attention in cognitive science as it deserves. There is clearly a gap that must be bridged between current education systems, in which the use of multiple external representations may actually hamper learning and expert performance, which is characterized by the ability to shift among different internal representations of a situation (Markman, 2000).

3. Learner-Centered Design

Learner-centered tools address the conceptual distance that lies between the audience of learners and the work practice in which they are engaged. Quintana, C et al. (2001) suggested that by developing scaffolding strategies informed by constructivist learning theories, a learner-centered tool incorporates support that allows a learner to participate in a new work practice to develop an understanding of the practice. The use of constructivist and social constructivist theories in learner-centered design (LCD) differs from the past efforts to design software tools for learning. Computer-aided instruction (CAI, based on behaviorist principles) and intelligent-tutoring system (ITS, based on information processing psychology) have attempted to package educational components that can train learners. Many CAI systems use a more information-transfer, passive model of learning. And ITS approaches can be less useful in loosely structured, wide-ranging work practices. While useful in more constrained fields like geometry and algebra, ITS tools can be difficult to implement for less constrained work activity. The key is that learner-centered tools should scaffold active participation by the learner in new work practices of all kinds.

4. Stimulative Design Computing Learning & Teaching Environment

Design computing is an interdisciplinary field that centers on the intersection of design, computer science, logic and linguistics. These multidimensional faces of this new paradigm necessitate a mutual understanding of computing and design as a whole, which are usually conceived as separated stages of computer-mediated design, i.e., Computer application and computing considered as the production force and design considered as the product for presentation. Bliss et al. (1999) addressed the nature of the influence of technology on learning, focusing especially on information technology and how it can influence any type of learning situation. Eastman (2001) has put forward the importance of cognition in design education. By means of the Vygotskian Inspired System (VIS), Luckin (1999) discussed the stimulative characteristics of learning with and by machines in assisting child-computer collaboration in the zone of proximal development.

Inspired by the above notions of computer-mediated collaborative learning, multiple representations and learner-centered design, in this research, interactions among artificial agents and human being in design computing learning is considered. In this paper the use of the above notions in stimulative design computing learning and teaching environment is explored.

5. Agent and Prototype Development

As shown in figure 1, an interactive and real-time design computing learning system agent model is constructed. By interactively manipulating the various attributes, codes, parameters and digital media contents in a distributed system, learners are gradually exposed to the mutual relationship of representational digital multimedia and the underlining programming syntax. The system combined the interactive coding verification capacity of visual software development utility such as Visual Studio and interactive multimedia manipulability of authoring utility such as Flash, with the introduction of 3D real-time simulation capacity for collaborative design computer learning in a distributed environment.

The first human computer interface (HCI) tools were developed using programming languages that were interpreted: Smalltalk (Tesler, L., [4) and then Dlisp (Teitelman, W., [5]). The interpreted language enables the developer to rapidly prototype different user interface ideas and immediately makes changes, which provides fast turnaround. With the rise of C and C++, most user interface development migrated to compiled languages and these capabilities were lost (Myers, B., Hudson, S.E. and Pausch, R., [6]). Researchers have investigated ways to bring back these advantages, resulting in the scripting languages such as tcl/tk (Ousterhout, J.K., [7]), Python (Lutz, M., [8]) and Perl (Wall, L. and Schwartz, R.L. [9]). Nowadays, popular scripting languages such as JavaScript and VBScipt are providing interpreted capabilities. Combining scripting capabilities with components and an interface builder had proven to be a particularly powerful approach.

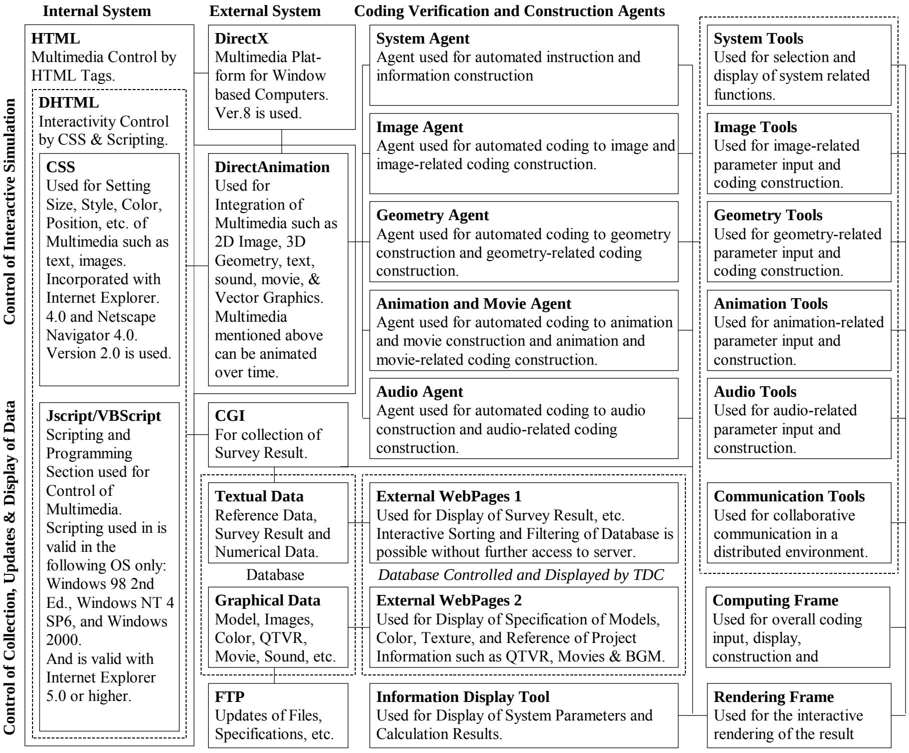
Internal System
External System
Coding Verification and Construction Agents
Control of Interactive Simulation
Control of Collection, Updates & Display of Data
HTML
Multimedia Control by HTML Tags.
DHTML
Interactivity Control by CSS & Scripting.
CSS
Used for Setting Size, Style, Color, Position, etc. of Multimedia such as text, images. Incorporated with Internet Explorer. 4.0 and Netscape Navigator 4.0. Version 2.0 is used.
Jscript/VBScript
Scripting and Programming Section used for Control of Multimedia. Scripting used in is valid in the following OS only: Windows 98 2nd Ed., Windows NT 4 SP6, and Windows 2000. And is valid with Internet Explorer 5.0 or higher.
DirectX
Multimedia Plat-form for Window based Computers. Ver.8 is used.
DirectAnimation
Used for Integration of Multimedia such as 2D Image, 3D Geometry, text, sound, movie, & Vector Graphics. Multimedia mentioned above can be animated over time.
CGI
For collection of Survey Result.
Textual Data
Reference Data, Survey Result and Numerical Data.
Database
Graphical Data
Model, Images, Color, QTVR, Movie, Sound, etc.
FTP
Updates of Files, Specifications, etc.
System Agent
Agent used for automated instruction and information construction
Image Agent
Agent used for automated coding to image and image-related coding construction.
Geometry Agent
Agent used for automated coding to geometry construction and geometry-related coding construction.
Animation and Movie Agent
Agent used for automated coding to animation and movie construction and animation and movie-related coding construction.
Audio Agent
Agent used for automated coding to audio construction and audio-related coding construction.
External WebPages 1
Used for Display of Survey Result, etc. Interactive Sorting and Filtering of Database is possible without further access to server.
Database Controlled and Displayed by TDC
External WebPages 2
Used for Display of Specification of Models, Color, Texture, and Reference of Project Information such as QTVR, Movies & BGM.
Information Display Tool
Used for Display of System Parameters and Calculation Results.
System Tools
Used for selection and display of system related functions.
Image Tools
Used for image-related parameter input and coding construction.
Geometry Tools
Used for geometry-related parameter input and coding construction.
Animation Tools
Used for animation-related parameter input and construction.
Audio Tools
Used for audio-related parameter input and construction.
Communication Tools
Used for collaborative communication in a distributed environment.
Computing Frame
Used for overall coding input, display, construction and
Rendering Frame
Used for the interactive rendering of the result

Figure 1. Prototype Development

Based on the use of JavaScript, VBScript and DirectX Scripting in combination with the DirectX multimedia platform in a web-based distributed environment, an interactive design computing learning system is constructed. Learners can interactively predict and evaluate the combined effect of digital multiple representations of multimedia contents (qualitative, semiotic, semantic) and programming codes (quantitative, mathematical presentation, syntax) real-time in a distributed environment. The aim is to expose the underlining mutual relationship of design and computing which are either intentionally hidden or simplified by coding automation functionalities of authoring software. This interface provides dynamic and intuitive access to programming and multimedia resources. Learners learn from their tentative trials on the combination of scripting, multimedia, parameters and attributes. The result of the design and coding can be discussed in a distributed environment, and the final coding can be saved and used in standalone basis for building up prospective system or presentational multimedia contents. The system is considered both a learning and production tool in the stream of design computing.

6. Conclusion

The goal of this research is to demonstrate the importance of multiple representations and scaffolding strategies in design computing education software development. This research is premised on the belief that stimulative design computing learning and teaching based on a multiple presentations and learner-centered design approach can lead to significant improvements in the effectiveness of design courses and to the future capabilities of practicing designers in architecture and the allied arts by means of digital multimedia and computing languages. This research is under progress.

Acknowledgements

The author would like to thank the research team of the Key Centre of Design Computing and Cognition, as well as my present and previous supervisors for past and ongoing discussion on many of the ideas of computer-mediated learning and teaching systems, multiple representation and learner-centered design in human computer interface development.

References

Bliss, J., Säljö, R. and Light, P (eds.): 1999, Learning Sites – Social and Technological Resources for Learning, Pergamon.

Boshuizen, H.P.A. and Schiff, H.J.M.: 1998, Problem solving with multiple representations by multiple and single agents: an analysis of the issues involved, in Van Someren, M. W., Reimann, P., Boshuizen, H.P.A. and De Jong, T. (ed.), Learning with Multiple Representations, Pergamon.

Dillenbourg, P. (ed.): 1998, Collaborative Learning: Cognitive and Computational Approaches, Pergamon.

Eastman, C., Newstetter, W. and McCracken, M. (eds.): 2001, Design Knowing and Learning: Cognition in Design Education, Elsevier.

Lucking, R.: 1999, Assisting child-computer collaboration in the zone of proximal development, in Bliss, J., Säljö, R. and Light, P (eds.), Learning Sites – Social and Technological Resources for Learning, Pergamon.

Lutz, M.: 1996, Programming Python, O'Reilly & Associates.

Markman, A.B.: 2000, British Journal of Educational Psychology, Vol.70 (1).

Myers, B., Hudson, S.E. and Pausch, R.: 2001, Past, Present, and Future of User Interface

Software Tools, in Carroll, J.M. (ed.), Human-Computer Interaction in the New Millennium, ACM.

Ousterhout, J.K.: 1991, An X11 toolkit based on the tcl language. Winter USENIX Technical Conference, 105-115.

Plomp, T. and Ely, D. P. (ed.): 1996, International Encyclopedia of Educational Technology.

Quintana, C., Carra, A., Krajcik, J. and Soloway, E.: 2001, Learner-Centered Design: Reflections and New Directions, in Carroll, J.M. (ed.), Human-Computer Interaction in the New Millennium, ACM.

Reifel, S. and Brown, M. H. (ed.): 2001, Early Education and Care, and Reconceptualizing Play. JAI.

Seel, N.M. and Dörr, G.: 1994, The supplantation of mental images through graphics: Instructional effects on spatial visualization skills of adults, in Schnotz, W. and Kulhavy, R. W. (eds.), Comprehension of Graphics, North-Holland.

Teitelman, W.: 1979, A display oriented programmer's assistant, International Journal of Man-Machine Studies, 11, 2, 157-187.

Tesler, L.: 1981, The Smalltalk environment, Byte Magazine, 6, 8, 90-147.

Van Someren, M. W., Reimann, P., Boshuizen, H.P.A. and De Jong, T. (ed.): 1998, Learning with Multiple Representations, Pergamon.

Wall, L., and Schwartz, R.L.: 1992, Programming perl. Sebastopol, CA: O'Reilly & Associates.