

Redesigning a Web-Conferencing Environment to Scaffold Computing Students' Creative Design Processes

Matt Bower

Department of Education, Macquarie University, Australia // Tel: +61 2 9850 8626 // matt.bower@mq.edu.au

ABSTRACT

Based on a three-semester design research study, this paper argues the need to redesign online learning environments to better support the representation and sharing of factual, procedural, and conceptual knowledge in order for students to develop their design capabilities. A web-conferencing environment is redesigned so that the modalities facilitate more socio-constructivist pedagogies whereby students co-construct knowledge and share their design thinking. The design-based research methodology explicates how redesign not only allowed more effective student representation and sharing of the different levels of knowledge required for abstraction to occur, but also enabled better teacher assessment and, hence, remediation. On the basis of the emerging findings of the study a set of principles for designing web-conferencing (or other online) environments to support creative design learning is proposed.

Keywords

Creative design, Web-conferencing, Technology, Learning design, Design-based research, Interface design

Introduction

Modern technologies provide teachers with new opportunities to create engaging and effective learning (Oliver, Harper, Wills, Agostinho, & Hedberg, 2007). Contemporary web-conferencing systems such as Adobe Connect (Adobe Systems Inc., 2010), Wimba Classroom (Wimba Inc., 2010) and Elluminate Live (Elluminate Inc., 2010) allow rich-media tools to be integrated, offering novel possibilities for instantiating synchronous online learning. Voice-over IP, text chat, whiteboards, screen-sharing, communal note areas, and so on provide a powerful suite of tools with which to present information, model processes, and share concepts.

Computer programming is a design-based field that depends upon the capacity of programmers to work creatively. It is a unique domain in that it is highly structured in terms of the underlying syntax and semantics, yet highly creative in terms of the solutions that can be produced. Based on a three-semester design-based research study of teaching computer programming in a web-conferencing environment, this paper reports on how the pedagogies, tasks, and interfaces utilized in online environments can be redesigned to effectively support students' creative design thinking and sharing.

Literature review

Creative design

Design can be defined as devising “courses of actions aimed at changing existing situations into preferred ones” (Simon, 1996, p. 111). Löwgren (1995) distinguishes between creative design and engineering design, with creative design being a more personal and unpredictable process resulting in the creation of many parallel ideas and concepts, whereas engineering design involves finding solutions to precisely defined problems. Importantly, in an attempt to dispel negative connotations associated with creative design processes and promote its intellectual rigor, Wolf, Rode, Sussman, & Kellogg (2006) point out that rather than being diametrically opposed, engineering design often involves elements of divergent and artistic production while creative design often contains structured practice and scientific reflection. That is to say, all design involves elements of creativity and science.

The way in which people represent and exchange design information has been proposed to critically affect the success of collaborative design processes. Carroll, Thomas & Malhotra (1980) found a significantly improved performance in a temporal design task when a visual representation was used. The arrangement of physical spaces has been observed to affect the success of design teams (either positively by providing natural ways to share created knowledge, or negatively by physically distancing designers), causing researchers to conclude that online

collaborative systems needs to facilitate effective switching between communicating and acting (Leiva-Lobos, De Michelis, & Covarrubias, 1997).

Based on an analysis of car design software and practices, Tano (2003) observed that different types of representations are more or less appropriate for different types of creative design thinking. Abstract sketches were found to be more appropriate for reflective cognition, and high reality renderings were more suitable for experimental cognition (Tano, 2003). In another study, the conception phase and the deliverable preparation phases of a creative design process were performed in entirely different ways (Leiva-Lobos et al., 1997). When considered in the light of designing online learning environments to support design processes, different interface designs may be more or less appropriate based on the phase of development and the type of thinking being engaged.

This design-based research study explains how the pedagogical approach, type of task, and interface of an online collaborative learning environment was redesigned in order to better support creative design processes.

Pedagogies for supporting creative design

There is a variety of pedagogies for supporting creative design ranging from transmissive to more student-centred approaches. Expert modelling is an instructional technique whereby the teacher demonstrates a to-be-learned process and uses explanation to offer students a “cognitive apprenticeship” (Collins, Brown, & Holum, 1991). This allows teachers to impart subject-matter knowledge, thought processes, problem-solving techniques, and a range of other capabilities that underpin creative design. Instructional approaches such as this are considered by some to be more appropriate when students are yet to form understandings about a particular topic (Magliaro, Lockee, & Burton, 2005). However these more transmissive approaches generally do not take maximum advantage of the benefits derived from more student-centred pedagogies, which include the active engagement of students, support from peers, and the ability to socially construct meaning (Hedberg, 2003; Jonassen, 2000; Land & Hannafin, 2000).

Computer-supported collaborative learning (CSCL) is sub-discipline of computer-mediated communication (CMC), which focuses on how online technologies can be used to design learning experiences whereby students share ideas and negotiate meaning (Jonassen, Lee, Yang, & Laffey, 2005). When groups are provided with collaborative technologies rather than being the recipients of online instruction, they have the capacity to co-construct meaning and distribute cognition between them (Hollan, Hutchins, & Kirsh, 2000; Wilson & Myers, 2000). Such constructionist approaches are argued to improve learning by virtue of engaging participants in personally meaningful productive pursuits over which they exercise a large degree of control (Willett, 2007). The real-world, ill-defined, and complex nature of authentic tasks posed in collaborative learning environments is espoused to improve student motivation and help transfer learning to other contexts (Herrington, Oliver, & Reeves, 2002).

The synchronous and multimodal capacities of many contemporary online learning environments afford new possibilities for moving from instructional pedagogies towards more social and construction-based learning as a way of developing students’ design capabilities.

Types of design knowledge

In accordance with both cognitive science (Byrnes, 2001) and contemporary models of learning (Anderson & Krathwohl, 2001), creative design thinking can be seen to be based on a range of different levels of knowledge:

1. Factual creative design knowledge — discrete pieces of elementary information, required for people to design and solve problems within a discipline
2. Procedural creative design knowledge — the skills to perform design processes and execute design procedures
3. Conceptual creative design knowledge — interrelated representations of more complex knowledge forms, including schemas, categorization hierarchies, and explanations of design constructs.

All three of these levels of knowledge are both essential and inseparably interrelated for developing creative design thinking. An understanding of creative design concepts arises from performing creative design processes, which are based upon items of factual knowledge. For instance, in the field of computer science, an understanding of how to

design a drawing program depends in the ability to perform all the steps to write to the drawing canvas, which is based upon an understanding of the syntax of the code relating to plotting points on the screen.

The process of building from facts to more abstract concepts based on performing processes has been described by Ahanori (2000) in the Actions-Process-Object model (see Figure 1).

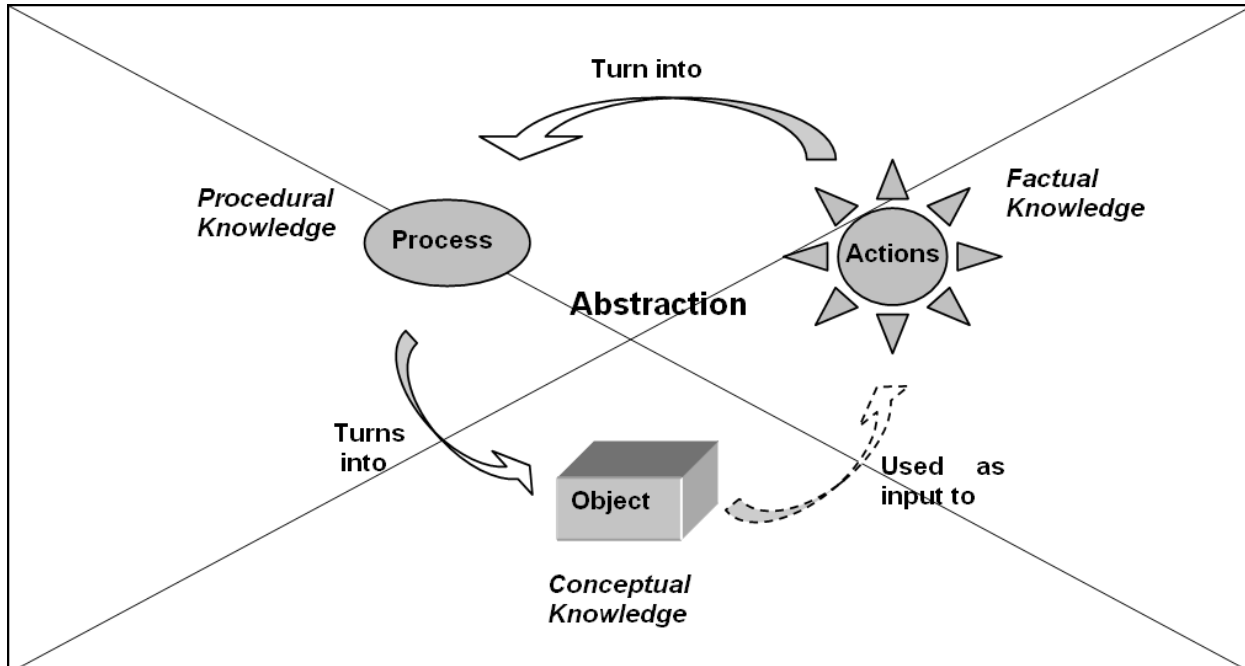


Figure 1. Ahanori's (2000) Actions-Process-Object model of abstraction and its relationship to factual, procedural and conceptual knowledge

As people perform sequences of actions as part of problem-solving processes, they form objects of knowledge that can then be used as inputs into other contexts (Ahanori, 2000). By writing computing code that draws points on a canvas to form an image, people can then abstract the concept of “drawing,” which can then be used to solve larger problems. It is this process of abstracting principles that is critical to effective learning because it allows concepts to be applied to other situations (Hazzan, 2003). Based on the Action-Process-Object model of abstraction, it is necessary to iteratively develop students’ factual, procedural, and conceptual design knowledge so that they may be able to apply their skills creatively in other contexts.

Multimedia learning effects

Several “multimedia learning principles” (Mayer, 2005) inform how online interfaces can be redesigned to better support creative design learning. Examples include:

- The “multimedia effect” — people learn more effectively from words and pictures than from words alone (Fletcher & Tobias, 2005)
- The “modality effect” — presenting some content in visual mode and other parts in auditory mode can lead to more effective learning than using text to supplement visual information (Low & Sweller, 2005)
- The “split-attention effect” — people learn more effectively when related words and pictures are located spatially or temporally close to one another (Ayres & Sweller, 2005)
- Symbol System Theory — matching the modality to the nature of the information being communicated can reduce the level of elaboration and recoding required for learner comprehension (Salomon, 1994).

The explanation for these effects is based upon cognitive science, in particular:

- There is a limit to the amount of information people can process at any one time (Cognitive Load Theory, see van Merriënboer & Ayres, 2005).

- People have the ability to process visual and auditory information somewhat simultaneously (dual-processing capabilities, see Pavio, 1986).

The way in which modalities are arranged in combinations or “clusters” has a considerable impact on the way information is interpreted (Baldry & Thibault, 2006). Designing environments for the development of creative design capabilities in part depends on ensuring that the way in which modalities are combined enables effective communication of processes and concepts.

Creative design and computer science

Computer programming is a design science. Depending on the level of expertise of the programmer, different approaches to design may be employed, from more procedural (bottom-up) approaches by novices to more functional (top-down) approaches by experts (Rist, 1995). For students to learn how to successfully design programs, they must first understand how to perform lower-order programming processes such as writing syntactically correct code, detecting and correcting flaws, predicting the effect of program segments, and applying programming concepts (Bower, 2008a; Fuller, et al., 2007).

Designing effective computer programs relies on the capacity of programmers to abstract their knowledge beyond the local, concrete representations they hold (Robins, Roundtree, & Roundtree, 2003). Kurland, Pea, Clement, & Mawby (1989) propose that the supporting computing students to abstract concepts is the key to their effective development. There is some evidence to show that design concepts can to some extent be learnt independently of the syntax and semantics that actualize them, although the capacity to put them into practice is necessary to be able to construct creative solutions (Fay & Mayer, 1994). Understanding design patterns underpins extensible and flexible programming (Gamma, Helm, Johnson, & Vlissides, 1998), and being able to design creatively using such abstractions is arguably the essence of computer science. Thus supporting students to abstract programming concepts is critical to developing their design capabilities.

Developing and assessing creative design thinking

In this study, Biggs & Collis’ (1982) Structure of Observed Learning Outcome (SOLO) taxonomy is used to measure the structure and sophistication of student representations of their mental models. Depending on the type of pedagogy applied, the level of knowledge involved in the task, and the nature of the interface provided, students present creative design thinking at different levels of the SOLO taxonomy:

- Prestructural — no correct response item in relation to a problem
- Unistructural — a single correct response item in relation to a problem
- Multistructural — several correct response items but not an entire set and not entirely interrelated with one another
- Relational — an entire set of fully interrelated response items in relation to a problem
- Extended abstract — a complete set of fully interrelated response items that are also interrelated to other information beyond the bounds of the concept or process being considered.

With respect to creative design, a “response item” could be considered either a piece of design knowledge or a step in a creative design process. If teachers do not provide students with the opportunity to represent creative design processes and concepts at the level of structure being addressed then students cannot practise or develop their creative design capabilities, and the teacher cannot adequately assess student understanding. In order for this to happen in online environments, the pedagogies, tasks, and tools must be selected to support representation at the level of structure being shared and assessed.

Method

This study applied a design-based research methodology across three semesters of an online introductory computer programming course to analyze how a web-conference-based learning environment could be redesigned to better

facilitate co-constructed design and learning. Design-based research involves iterative cycles of problem analysis, theory-based solution development, evaluation in real-world settings, and the development of situated design principles (Reeves & Hedberg, 2003). The difference between predictive and design research approaches is summarized in Figure 2.

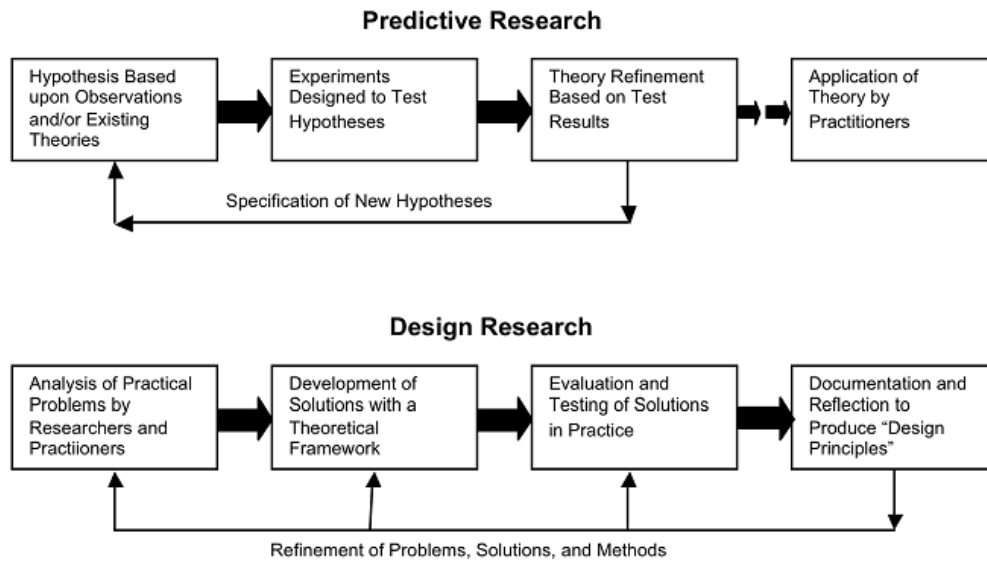


Figure 2. Difference between predictive and design research approaches (Reeves & Hedberg, 2003)

Design-based research was selected in this study for its capacity to cater to the real-life complex educational setting, to integrate theoretical principles with cycles of in-situ iterative reflection, and to shape grounded yet generalisable design principles (Cobb, Confrey, diSessa, Lehrer, & Schauble, 2004; Collins, Joseph, & Bielaczyc, 2004). Design-based research enables the use of any and all types of data to arrive at effective designs (Gorard, Roberts, & Taylor, 2004; Wilson, 2004), which allowed the teacher’s observations as well as students’ interactions, contributions, discourse, and feedback to inform the redesign of the environment.

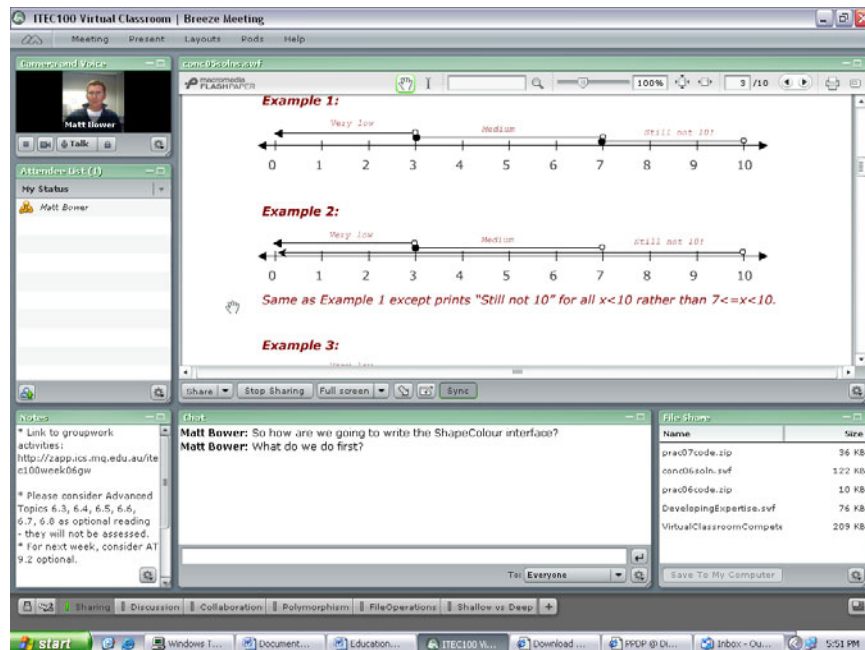


Figure 3. An Adobe Connect Meeting interface, showing (clockwise from top left) Camera and Voice, Document share, File Share, Chat, Notes and Attendee List Pods

The web-conferencing platform

The Adobe Connect Meeting platform (Adobe Systems Inc., 2009) is a web-conferencing system that provides the flexibility to tailor the interface to meet the requirements of the learning episode (see Figure 3).

Designers can select from a range of tools, including those to display PowerPoint presentations, broadcast webcam and voice, screen-share, exchange text chat and files, vote, write shared notes, and collaboratively draw on a whiteboard. Each of these tools (or pods) can be instantly resized, dragged-and-dropped, created or deleted, allowing the designer ultimate flexibility in the tools that are provided and the way that they are arranged. The platform also allows several layouts to be pre-designed within one room and all meetings to be recorded.

The context

This research focused specifically on collaborations in the two-hour weekly online classes for an introductory software development subject at Macquarie University. The subject aimed to teach students the basic skills of computer programming as well as develop their capacity to design effective and creative solutions using computing constructs. The unit of study covered basic programming syntax and semantics, objects and classes, polymorphism and inheritance, applets and GUIs, arrays and ArrayLists, as well as error handling and file operations. The language used to develop these capabilities was Java (Sun Microsystems Inc, 2010).

Students who undertook the introductory programming subject were graduate students from a discipline other than computing who wished to extend their IT knowledge and skills. There were 26 students who enrolled across the three semesters, of which 20 completed the unit. Of these 20 students, ten were enrolled in 2005, Semester 2; seven in 2006, Semester 1; and three in 2006, Semester 2. Of the 26 students, 9 were female and 17 were male.

The method

The approaches to redesigning the learning environment across the three iterations can be summarized as follows:

- **Iteration 1:** The approach to teaching was predominantly transmission based, including long periods of teacher explanation and presentation of solutions relating to tutorial and practical exercises. In some cases a degree of interaction was afforded using audio to ask students questions and having students respond using text chat. The default layouts of the web-conferencing platform, or minor variations thereof, were applied. This iteration offered a baseline for the design-based research analysis.
- **Iteration 2:** The learning activities and web-conferencing environment were redesigned to facilitate student participation in creative design processes. Tasks required students to contribute their design ideas and create computer programs. There was greater use of screen-sharing and Note-pod tools to enable students to collaboratively perform programming processes, though text chat was used to mediate the majority of discourse.
- **Iteration 3:** Greater emphasis was placed on students sharing their conceptual understanding of program designs, either before, during or after collaborative programming processes were performed. The web-conferencing environment was redesigned, often spontaneously, to incorporate whiteboards so that students could represent their mental models in visual form. As well, pervasive use of audio by students was used to facilitate collaboration.

Recordings of these lessons were captured and analysed ex-post facto to identify factors that affected the success of the learning episodes. These reflections on the process of enactment were recorded in a reflective journal and added to the project database along with other relevant data (such as student programs and correspondence that occurred outside the virtual classroom). Both successes and failures were documented, with the failures seen as making an important contribution to understanding teaching and learning in the web-conferencing environment.

The observations contained in the reflective journal notes along with lesson artefacts and student feedback were then used as a basis for redesign, in accordance with design-based research processes (Gorard, Roberts, & Taylor, 2004; Wilson, 2004). The influence of both tactical (within semester) and strategic (across semester) redesigns on collaboration and learning were added to the project database through their inclusion in the reflective journal. Validation of effects occurred by repeated observation either within or across iterations, or both. Data triangulation between recording of lessons, learning artefacts, and student feedback was also used to establish the effectiveness or

otherwise of approaches. Applied principles for teaching and learning in web-conferencing environment were then derived by distilling repeated effects as they applied to supporting and developing creative design capabilities.

Results

Descriptions of key observations and particular learning episodes have been provided to illustrate the design research process. Vignettes are used to depict critical incidents and effects with relation to the pedagogies, types of knowledge being addressed, and the interface designs applied. While it is not possible to include all data and analysis in this paper, a more complete portrayal of the study is available online (Bower, 2008b).

Iteration 1

In the first iteration, standard interface layouts, or minor variations thereof, were used. The two main pedagogical strategies used to develop students' design capabilities were both teacher dominated: broadcasting exemplar solutions and demonstrating programming processes. An example of broadcasting exemplar solutions is illustrated in Vignette 1.

Vignette 1

The screenshot in Figure 1 illustrates how the teacher broadcast the Week 1 tutorial solutions. The pedagogy was transmissive, the task involved predominantly factual knowledge, and the interface only supported text-chat contributions by students.

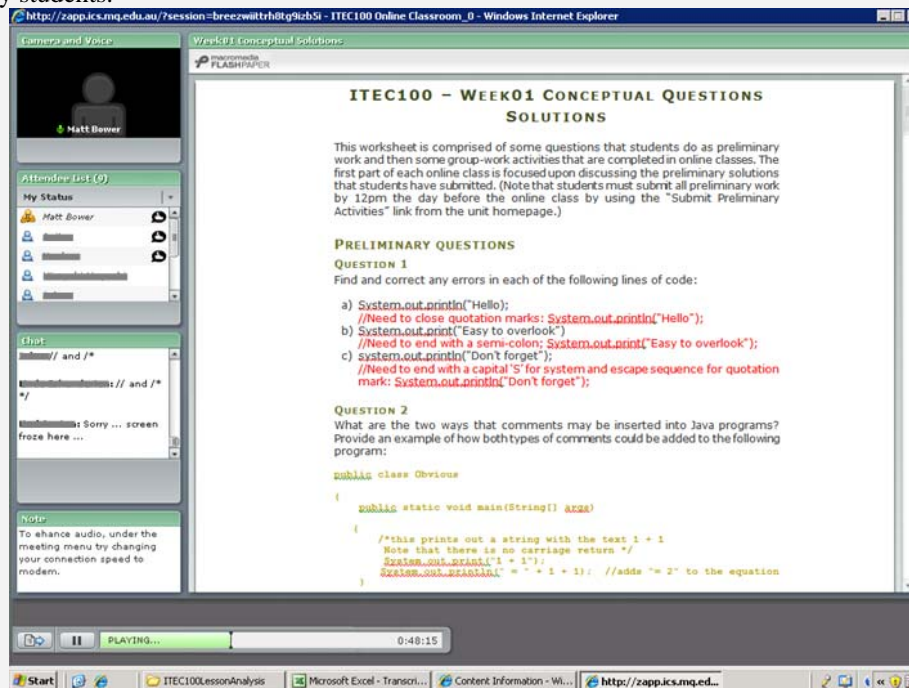


Figure 1. Teacher broadcasting exemplars to students using the standard web-conferencing interface

The teacher-centred pedagogy restricted students' ability to practise and represent their thinking. The task requiring only factual responses to unauthentic text-book style questions resulted in closed responses. The use of text chat inhibited the extent to which information could be interrelated or extended contributions could be made. At best, the text chat allowed a multistructural understanding to be demonstrated by virtue of students' contributing multiple pieces of information relating to the concept. It could not allow a relational understanding to be shared since students did not interrelate all items of knowledge in the same way that they would have if they were applying the information creatively as part of a problem-solving process. The teacher-centred pedagogy, the limited task, and the interface supporting only text-chat contributions did not support the expression or development of creative design thinking.

In Iteration 1, a teacher-dominated “modelling” approach was also applied in practical programming activities. This is exemplified in Vignette 2.

Vignette 2

Figure 2 shows how the teacher modelled programming approaches by broadcasting their screen and used audio to provide insight into underlying thought processes being performed. While the task was more authentic, students could only make text-chat suggestions about how to design the computer program and ultimate control over the process resided with the teacher.

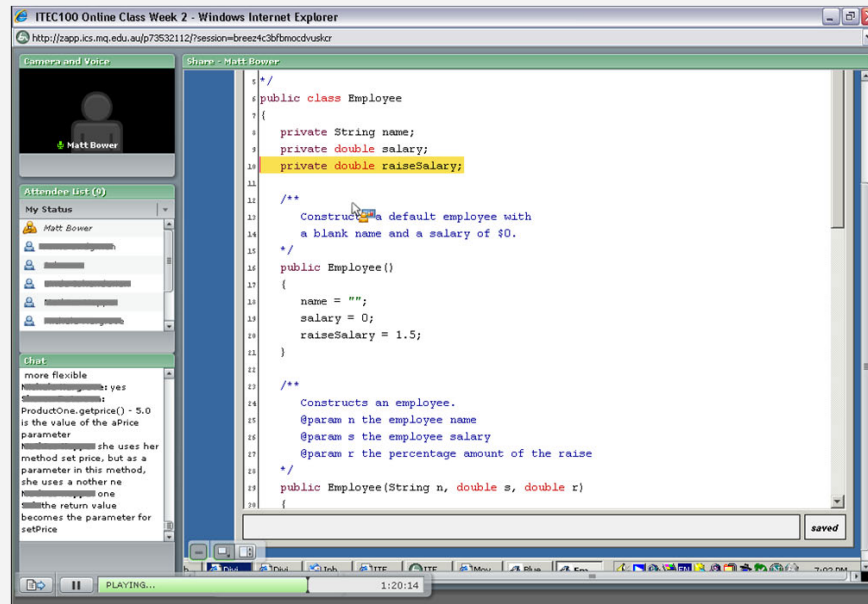


Figure 2. Basic screen-sharing approach to model programming processes

This approach was useful during the early stages of the course when students had little or no design-process knowledge. The screen-sharing and teacher audio enable the teacher to offer students a cognitive apprenticeship on how to edit, compile, and debug program code. However the approach once again only allowed students to demonstrate a multistructural understanding by virtue of contributing several text-chat comments. The approach did not allow a relational understanding to be evidenced because the teacher was completing procedural aspects and intermediary steps in the design process.

The screen-sharing provided a modality that was able to dynamically represent the process information being shared, thus representing information in a “cognitively efficient” form (in accordance with Symbol System Theory, Salomon, 1994). However the teacher was the major contributor of creative-process information.

Iteration 2

Based on observations from Iteration 1 indicating low levels of student contribution and teacher-dominated lessons, the learning environment was redesigned in Iteration 2 to engage more student-centred learning. This included groupwork activities where students were required to design solutions to problems in small teams. For instance, Vignette 3 illustrates how students used desktop-broadcasting to share process knowledge about how to write programs.

Vignette 3

For this activity students were divided into groupwork rooms and asked design a program that created cylindrical TinCan objects, which could return their own volume. The student-centred pedagogy and the authentic task enabled students to demonstrate their creative design capabilities.

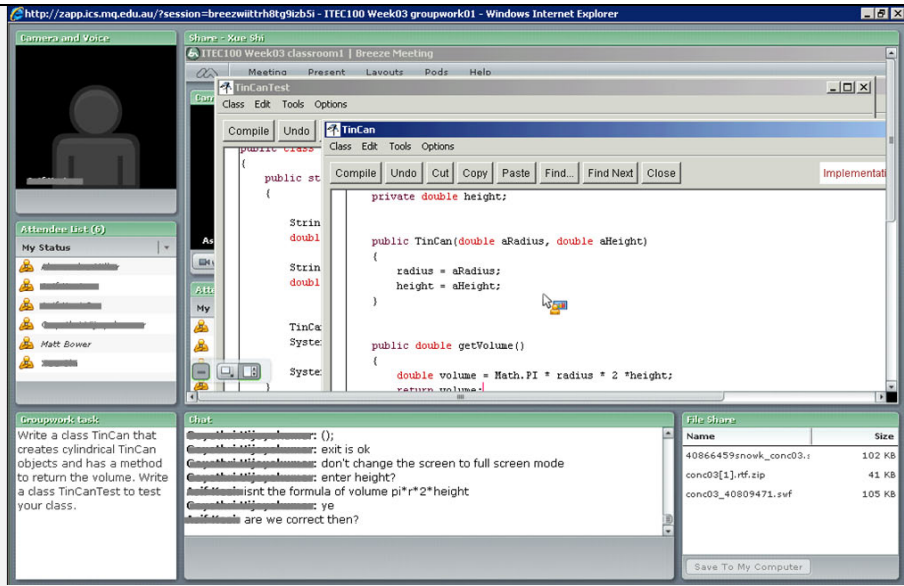


Figure 3. Students using screen-sharing to perform group programming

Figure 3 shows how one student was able to broadcast their desktop and other students could contribute suggestions using text chat. This time the teacher could assess a relational understanding by virtue of students being able to correctly perform and sequence all steps of the design process. However in this episode, the people broadcasting their screen struggled to communicate with the rest of the group because using text-chat collaboration rather than audio resulted in “split-attention” (Ayres & Sweller, 2005).

In Iteration 2, the capacity to utilize multiple Note-pod areas afforded the potential to create more cognitively and collaboratively efficient interface designs (as exemplified in Vignette 4).

Vignette 4

In this exercise, students were divided into groupwork rooms and required to merge a “circle resize” program with a “circle re-centre” program, combining functionality and removing any redundancies. The web-conferencing interface was redesigned so that the resize and re-centre programs were displayed in Note pods, and a third Note-pod column was provided for students to write their combined program (see Figure 4).

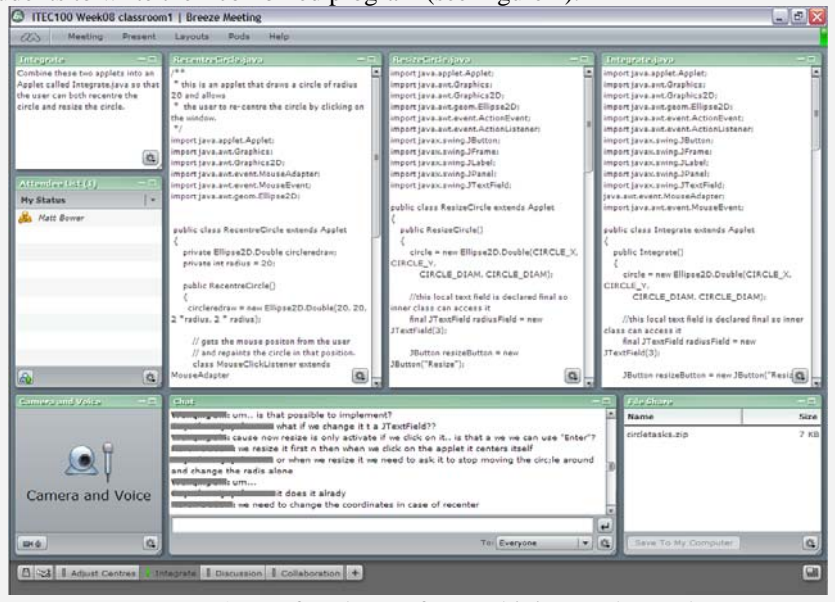


Figure 4. Interface layout for combining applets task

The student-centred pedagogy and authentic task allowed students to demonstrate and share their creative-process thinking. The layout integrating all the relevant information in the one interface, overcame the need to navigate back and forth between program files. The interface supported high levels of student contribution and was available for the teacher to review for diagnostic purposes. For instance, the teacher was able to determine that group 1 was unsure about how to solve this problem, making some progress but not understanding where some of the code from `RecentCircle` should be inserted into `ResizeCircle`. On the other hand, inspection of the group 2 virtual classroom revealed a complete (relational) understanding of how the program should be designed. However, the fact that all communication was mediated using text meant that it was difficult for everyone to follow who was making contributions, and visual concepts relating to the design were not able to be shared.

Iteration 3

Based on issues identified in the previous semester, Iteration 3 was characterized by greater use of whiteboards to facilitate the exchange of conceptual creative design knowledge. As well, audio was pervasively used to facilitate more efficient groupwork collaboration.

Vignette 5 demonstrates how a whiteboard was utilized to strengthen students' conceptual design knowledge.

Vignette 5

In response to student difficulties with nested array program design, a whiteboard was used to support conceptualisation of program operation. With the guidance of the teacher, students dynamically represent the state of variables and arrays while they stepped through the program (see Figure 5).

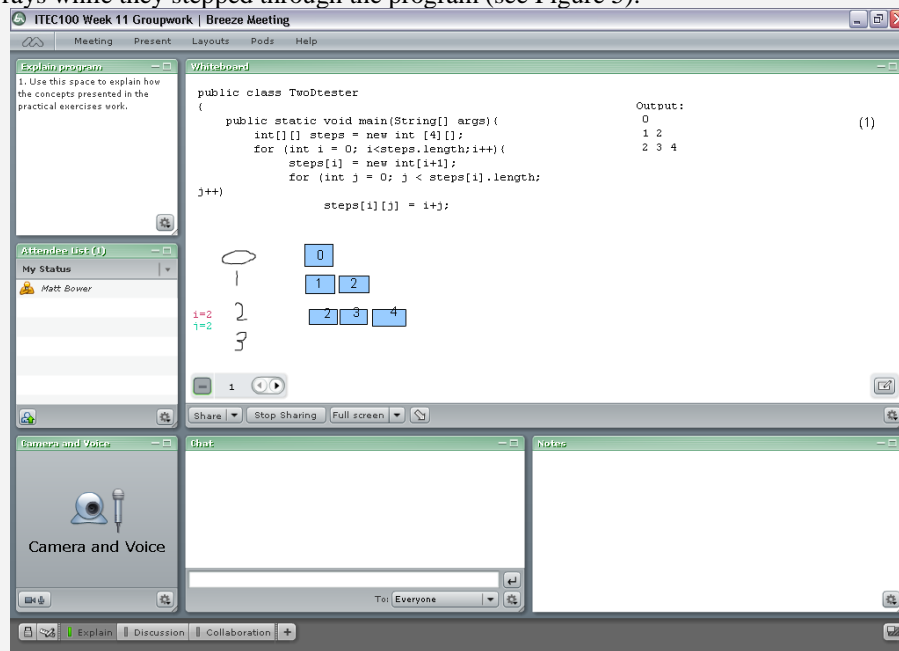


Figure 5. Using a whiteboard to support guided representation of a design concept

The pedagogy required students to represent their understanding in the solution space. The task, while less authentic in nature, directly responded to student misconceptions. Integrating a whiteboard into the interface allowed students to represent their understanding of the concept in a dynamic way so that the level of student comprehension was immediately evident to the teacher. Student comments indicated that communicating aspects of the concept visually as well as in words enabled them to form a more complete understanding, in accordance with the multimedia effect (Fletcher & Tobias, 2005). The contribution-based pedagogy and the interface supporting the sharing of conceptual knowledge had enhanced their understanding of the design pattern.

Audio was used pervasively throughout this iteration in an attempt to support more efficient collaboration. Audio was observed to allow student groups to contribute more discourse in less time than text chat, and those contributions could be made with greater ease. Using audio in small group situations enabled easier coordination of activity because students could contribute to the whiteboard or Note pods at the same time as they were speaking (which was not possible for text chat). For example, in Vignette 5 students were able to contribute to the whiteboard at the same time as they discussed the design concepts. Utilizing audio for student-centred tasks also reduced split-attention caused by monitoring and contributing to two visual channels at once (Note pod and text chat or whiteboard and text chat), thus allowing people's dual processing capabilities to be utilized (modality effect, Low & Sweller, 2005).

Iteration 3 was characterized by spontaneous redesign of interfaces to support the changing collaborative needs of the evolving conversation. This is exemplified in Vignette 6.

Vignette 6

For this activity students were required to design a flexible program that allowed multiple copies of an image to be drawn on a canvas. The pedagogy was student-centred in so far as they were responsible for constructing the solution. The authentic task required students to integrate their conceptual knowledge of object reuse with their procedural knowledge of how to write programs. A purpose built web-conferencing interface had been designed with the entire program code (main method and drawing object) displayed in the two Note pods along the right hand side of the interface so that split-attention was avoided (see Figure 6).

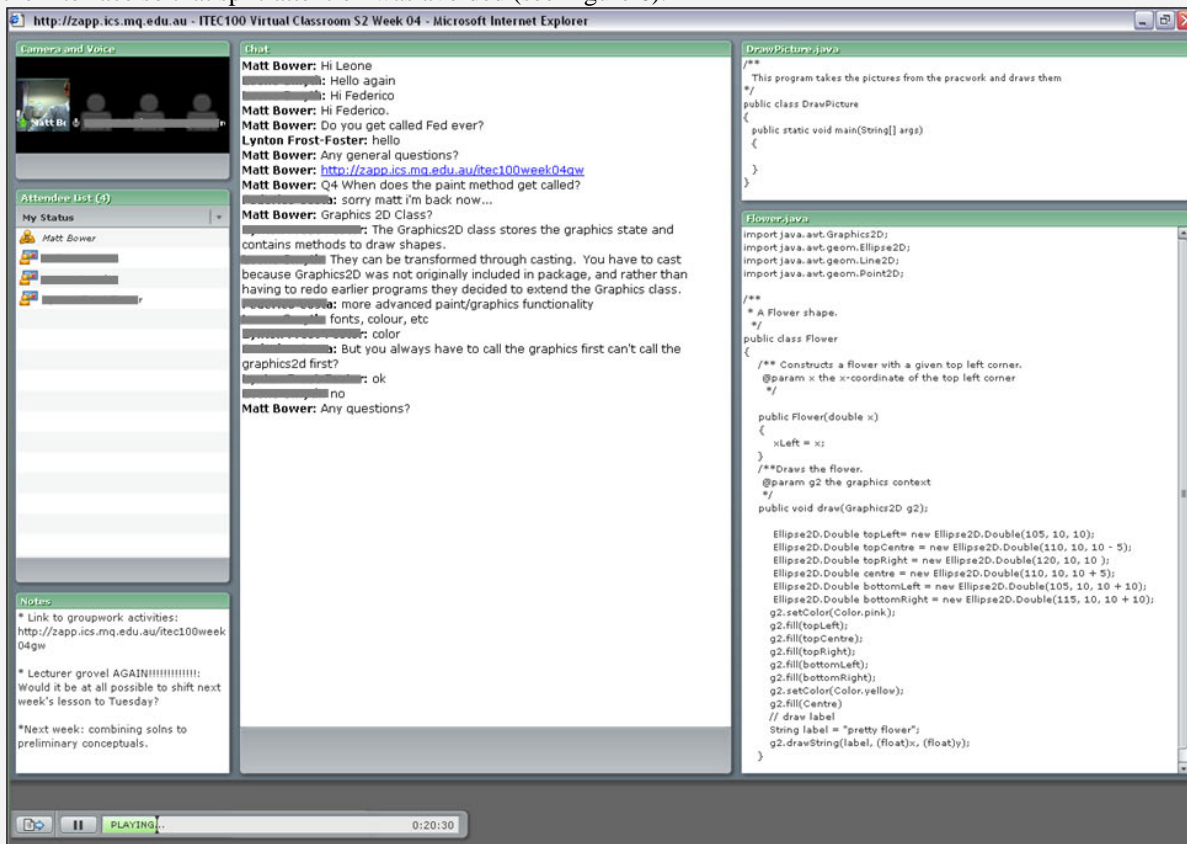


Figure 6. Initial design of the applet image drawing episode

Student progress and questions indicated that there was uncertainty about how the coordinate system of the image being drawn could be flexibly related to the coordinate system of the canvas. This prompted the teacher to adjust the web-conferencing interface mid-episode so that a whiteboard could be used to represent conceptual information (see Figure 7).

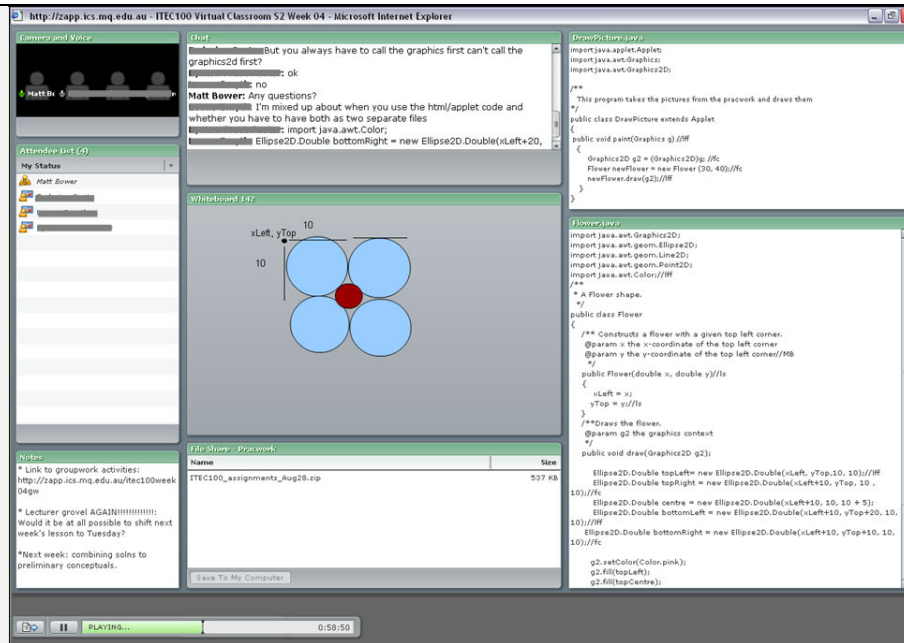


Figure 7. Dynamic adjustment to interface enabling effective conversational approach

As well as supporting the teacher's explanation, the whiteboard allowed students to represent their amended conceptions so that the teacher could gauge whether they had developed accurate mental models. The picture adjacent to the program code enabled students to relate the conceptual knowledge represented in the diagram to the programming process occurring in the Note pods. This supported the development of students' abstractions by relating phases of the Action-Process-Object cycle (Aharoni, 2000). Students then collaboratively adjusted the flower program to correctly incorporate the provision of x and y coordinates in the object constructor. The teacher could immediately assess a relational understanding had been achieved. By improving the program design to be flexible rather than hard coding the image coordinates it was then possible to quickly write a program that could draw multiple copies of the image across the canvas (see Figure 8).

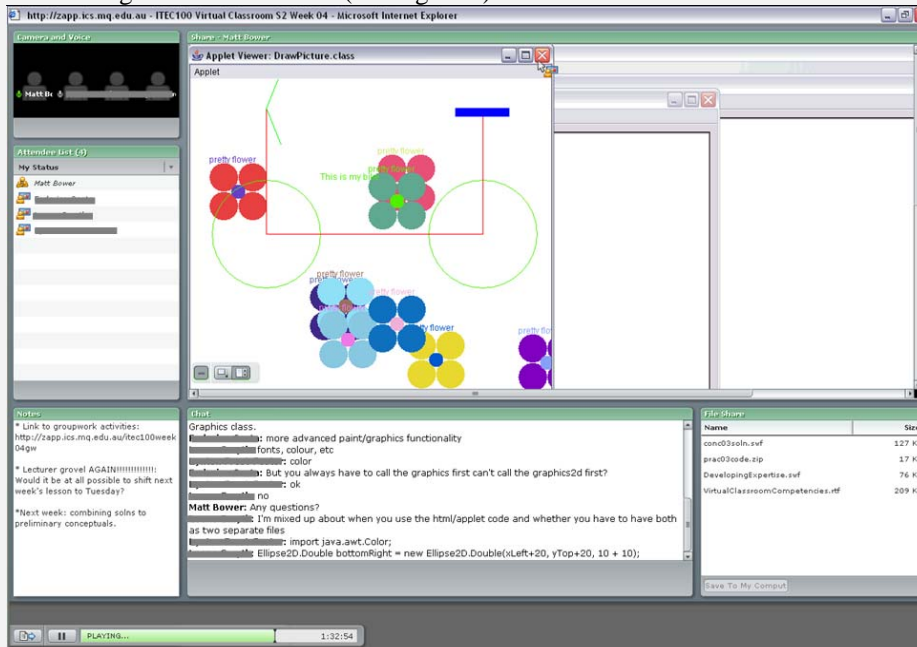


Figure 8. Dynamic adjustment to interface to represent design output

Note that this involved a transition to another interface design, this time a screen-sharing layout, in order to effectively represent the program output.

While the descriptions and examples above cannot fully explicate the design-based research process, observations, and results, it does provide an indication of the type of analysis that was conducted and the nature of the data collected. A more complete analysis is available online for readers who may be interested (Bower, 2008b).

Principles for scaffolding creative design learning in online environments

On the basis of observations drawn from the three iterations of the design-based research project, the following set of recommendations for developing students' creative design capabilities in online environments are proposed.

Adopt student-centred collaborative pedagogies to support the revelation and sharing of creative design thinking

Student-centred collaborative activities enable students to practise and share their creative design capabilities. For instance, by requiring students to collaboratively construct solutions to the “circle combine” and “flower paint” activities, the students are provided with the opportunity to actively apply their creative design knowledge and learning from one another. Because students more fully reveal their mental models through their solutions and conversations, the teacher can more accurately assess student understanding and provide remediation if required. For instance, in the circle combine activity, the teacher was able to gauge both the specific misconceptions held in one group and that the other group had achieved a relational understanding. In contrast, the more teacher-dominated approaches in Iteration 1 restricted the amount of contribution that students could make and thus the quality of response that the teacher could provide.

Apply authentic tasks in order to engage all levels of creative design thinking

More authentic tasks incorporate all stages of the Action-Process-Object cycle and thus support an integrated understanding of factual, procedural, and conceptual design knowledge. For instance the more authentic “tin can” and “flower paint” programming activities required students to integrate their factual (syntactic), procedural (programming processes), and conceptual (design pattern) knowledge. In contrast, the text-book style task in Vignette 1 at best allowed only factual knowledge to be shared and developed. Authentic tasks require students to work on a concrete problem, thus engaging their declarative and procedural knowledge. As students attempt to apply design patterns from other episodes to the current context and develop abstractions from their problem-solving attempts, conceptual thinking is engaged. Performing this in a public space allowed the abstraction process to be shared by all participants, allowing the evolving nature of students' underlying mental models to be revealed. As a result more authentic tasks also provide the teacher with greater insight into the accuracy of student schema and the form of remediation that may be required.

Select modalities that best suit the desired form of representation

Different representational possibilities are afforded by different modalities, and the modality of representation should be selected to match the collaborative and cognitive requirements of the learning episode. Following are some examples:

- Text chat — effective for simultaneous sharing of factual information among members of a large group of contributors (for instance, when several participants are making suggestions about the next step when writing a computer program)
- Audio — affords rapid contribution of extensive descriptions by one person (for instance, a team leader) or rapid turn-taking among a members of a small group of users (who may be collaboratively designing)
- Note pods — useful for organizing textual information among multiple users where sequencing, editing, copying, and deletion are required (for instance, collaborative authoring of a solution)
- Screen-sharing — suitable for sharing process-based information with relation to computing (such as performing a programming process)
- Whiteboard — effective for supporting shared representation and development of conceptual knowledge (for instance, drawing diagrams to represent the schematic design of a program)

Designing multimodal clusters according to multimedia learning principles to improve cognitive efficiency

Different modalities offer not only different individual possibilities for representing, but also different possibilities in combination as “multimodal clusters” (Baldry & Thibault, 2006). The design of effective multimodal clusters was observed to rely upon application of multimedia learning principles. Examples include the following:

- Using whiteboard diagrams to embellish audio explanations to support clearer formation of mental models (multimedia effect)
- Use of audio (rather than text chat) in combination with visual modalities (such as Note pods, whiteboards or screen-sharing) to leverage people’s dual-processing capabilities (modality effect)
- Placing related information such as program files and diagrams in close physical and temporal proximity (avoiding split-attention)
- Using modalities that accurately match the nature of information being presented, such as screen-broadcasting for programming processes, so that minimal recoding of that information is required (symbol system theory).

These principles provide strategies in response to claims by Carroll et al. (1980) and Leiva-Lobos et al. (1997) that the way in which people represent and exchange design information can critically affect the success of design endeavours.

It should be noted that while this study only reports on the design-based research findings elaborated in Bower (2008b), feedback from students verified the above principles in the following ways:

1. Active and authentic learning exercises better supported their learning (Bower, 2009).
2. The selection and arrangement of tools based on multimedia learning principles resulted in significantly superior interfaces (Bower & Hedberg, 2009).

Conclusion

The capacity to design creative solutions depends on the acquisition of underlying factual, process, and conceptual understanding. In order to develop students’ creative design capabilities it is critical that environments are designed to facilitate the effective contribution and exchange of knowledge at the level being addressed (factual, procedural, and conceptual). It is imperative that students are provided with the opportunity to represent their mental models in order for ideas to be more effectively shared with their peers and their level of understanding to be more accurately assessed.

Based on an analysis of teaching and learning computer programming in a web-conferencing environment this paper presents principles for designing online learning environments to better support creative production. These include applying student-centred and collaborative pedagogies to increase the amount of activity and contribution, utilising authentic tasks that provide students with the opportunity to integrate all levels of creative design thinking, selecting modalities that best support the type of collaboration and level of design knowledge being engaged, and clustering modalities in cognitively efficient forms in accordance with multimedia learning effects. Within this framework, ongoing adjustment of the environment may be required to support the emerging collaborative and cognitive requirements of a learning episode.

Multimedia and socio-constructivist learning principles can be used to support all stages of the action-object-process model of abstraction. In this study, the redesign of the learning environment involved a movement from instructive to student-centred pedagogies, whereby students performed collaborative design processes using screen-sharing and negotiating design concepts using shared whiteboards. Pervasive use of audio by students enabled more effective discourse around creative design artefacts. Supporting operation at and movement between the different levels of knowledge encourages abstraction of design capabilities and allows students to more easily apply their skills in other problem-solving situations. It is intended that the principles for redesigning online learning environments to facilitate abstraction of creative design concepts will support educators in developing the creative design capacities of their students in other educational settings.

References

- Adobe Systems Inc. (2010). *Acrobat Connect Professional [software]*, Retrieved October 29, 2010, from <http://www.adobe.com/products/acrobatconnectpro/>.
- Aharoni, D. (2000). Cogito, Ergo sum! cognitive processes of students dealing with data structures. *Proceedings of the thirty-first SIGCSE Technical Symposium on Computer Science Education*, 26–30.
- Anderson, L., & Krathwohl, D. (2001). *A taxonomy for learning, teaching and assessing: A revision of Bloom's taxonomy of educational objectives*, New York: Longman.
- Ayres, P., & Sweller, J. (2005). The split-attention principle in multimedia learning. In R. E. Mayer (Ed.), *The Cambridge Handbook of Multimedia Learning* (pp. 135–146), New York: Cambridge University Press.
- Baldry, A., & Thibault, P. J. (2006). *Multimodal transcription and text analysis*, London: Equinox Publishing Ltd.
- Biggs, J., & Collis, K. (1982). *Evaluating the Quality of Learning: the SOLO taxonomy*, London: Academic Press.
- Bower, M. (2008a). A taxonomy of task types in computing. *Proceedings of the 13th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE08)*, Madrid, Spain, 30 June–2 July (pp. 281–285), New York: ACM.
- Bower, M. (2008b). Designing for interactive and collaborative learning in a web-conferencing environment, *PhD thesis*, Retrieved October 29, 2010, from <http://hdl.handle.net/1959.14/26888>.
- Bower, M. (2009). Learning computing online: Key findings from students. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2009, Honolulu, HI*. (pp. 4166–4175), Chesapeake, VA: AACE.
- Bower, M., & Hedberg, J. (2009). Educational user interface design. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2009, Honolulu, HI*. (pp. 963–972), Chesapeake, VA: AACE.
- Byrnes, J. P. (2001). *Minds, Brains and Learning*, New York: The Guilford Press.
- Carroll, J., Thomas, J., & Malhotra, A. (1980). Presentation and representation in design problem solving. *British Journal of Psychology*, 71(1), 143–155.
- Cobb, P., Confrey, J., diSessa, A., Lehrer, R., & Schauble, L. (2004). Design Experiments in Educational Research. *Educational Researcher*, 32(1), 9–13.
- Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. *American Educator*, 6(11), 38–46.
- Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design Research: Theoretical and Methodological Issues. *Journal of the Learning Sciences*, 13(1), 15–42.
- Illuminate Inc. (2010). *Illuminate Live web collaboration software*, Retrieved 29th October, 2010, from <http://www.illuminate.com>
- Fay, A. L., & Mayer, R. E. (1994). Benefits of teaching design skills before teaching Logo computer programming: Evidence for syntax independent learning. *Journal of Educational Computing Research*, 11(3), 187–210.
- Fletcher, J. D., & Tobias, S. (2005). The multimedia principle. In R. E. Mayer (Ed.), *The Cambridge handbook of multimedia learning* (pp. 117–133), New York: Cambridge University Press.
- Fuller, U., Johnson, C. G., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jackova, J., et al. (2007). Developing a computer science-specific learning taxonomy. *ACM SIGCSE Bulletin*, 39(4), 152–170.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1998). *Design Patterns: Elements of reusable object-oriented software*, Reading, MA: Addison Wesley.
- Gorard, S., Roberts, K., & Taylor, C. (2004). What kind of creature is a design experiment? *British Educational Research Journal*, 30(4), 577–590.
- Hazzan, O. (2003). How students attempt to reduce abstraction in the learning of mathematics and in the learning of computer science. *Computer Science Education*, 13(2), 95–123.
- Hedberg, J. G. (2003). Ensuring quality e-learning: Creating engaging tasks. *Educational Media International*, 40(3/4), 175–186.
- Herrington, J., Oliver, R., & Reeves, T. C. (2003). Patterns of Engagement in Authentic Online Learning Environments. *Australian Journal of Educational Technology*, 19(1), 59–71.
- Hollan, J., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Transaction of Computer-Human Interaction*, 7(2), 174–196.

- Jonassen, D. H. (2000). *Computers as mindtools for schools: Engaging critical thinking (2nd Ed.)*, Upper Saddle River, NJ: Prentice Hall.
- Jonassen, D. H., Lee, C. B., Yang, C.-C., & Laffey, J. (2005). The collaboration principle in multimedia learning. In R. E. Mayer (Ed.), *The Cambridge handbook of multimedia learning* (pp. 247–270), New York: Cambridge University Press.
- Kurland, D., Pea, R., Clement, C., & Mawby, R. (1989). A Study of the development of programming ability and thinking skills in high school students. In E. Soloway & J. C. Spohrer (Eds.), *Studying the novice programmer* (pp. 83–112). Hillsdale, NJ: Lawrence Erlbaum.
- Land, S. M., & Hannafin, M. J. (2000). Student centred learning environments. In D. H. Jonassen & S. M. Land (Eds.), *Theoretical foundations of learning environments* (pp. 1–23), Mahwah, NJ: Lawrence Erlbaum Associates.
- Leiva-Lobos, E. P., De Michelis, G., & Covarrubias, E. (1997). *Augmenting and multiplying spaces for creative design*. Paper presented at the Conference on Supporting Group Work, Phoenix, AZ, USA.
- Low, R., & Sweller, J. (2005). The modality principle in multimedia learning. In R. E. Mayer (Ed.), *The Cambridge Handbook of Multimedia Learning* (pp. 147–158), New York: Cambridge University Press.
- Löwgren, J. (1995). Applying design methodology to software development. In *Proceedings of the 1st Conference on Designing Interactive Systems: Processes, Practices, Methods, & Techniques*, (pp. 87–95), New York: ACM Press.
- Magliaro, S. G., Lockee, B. B., & Burton, J. K. (2005). Direct instruction revisited: A key model for instructional technology. *Educational Technology Research & Development*, 53(4), 41–55.
- Mayer, R. E. (2005). Introduction to multimedia learning. In R. E. Mayer (Ed.), *The Cambridge handbook of multimedia learning* (pp. 1–17), New York: Cambridge University Press.
- Oliver, R., Harper, B., Wills, S., Agostinho, S., & Hedberg, J. (2007). Describing ICT-based learning designs that promote quality learning outcomes. In H. Beetham & R. Sharpe (Eds.), *Rethinking pedagogy for a digital age: Designing and delivering e-learning* (pp. 64–80), Oxford, UK: Routledge.
- Pavio, A. (1986). *Mental representations: A dual coding approach*, New York: Oxford University Press.
- Reeves, T. C., & Hedberg, J. (2003). *Interactive Learning System Evaluation*, Englewood Cliffs, NJ: Educational Technology Publications.
- Rist, R. S. (1995). Program Structure and Design. *Cognitive Science*, 19, 507–562.
- Robins, A., Roundtree, J., & Roundtree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.
- Salomon, G. (1994). *Interaction of media, cognition, and learning*, New Jersey: LEA.
- Simon, H. A. (1996). *The Sciences of the Artificial*, Cambridge, MA: MIT Press.
- Sun Microsystems Inc (2010). *Java Home Page*, Retrieved October, 29, 2010, from <http://www.java.com/>
- Tano, S. I. (2003). Design concept to promote both reflective and experimental cognition for creative design work. In *Conference on Human Factors in Computing Systems*, Ft. Lauderdale, FL, (pp. 1014–1015): ACM.
- van Merriënboer, J. J. G., & Ayres, P. (2005). Research on cognitive load theory and its design implications for e-learning. *Educational Technology Research & Development*, 53(3), 5–13.
- Willett, R. (2007). Technology, pedagogy and digital production: A case study of children learning new media skills. *Learning, Media and Technology*, 32(2), 167–181.
- Wilson, B. G. (2004). Designing e-learning environments for flexible activity and instruction. *Educational Technology Research & Development*, 52(4), 77–84.
- Wilson, B. G., & Myers, K. M. (2000). Situated cognition in theoretical and practical context. In D. H. Jonassen & S. M. Land (Eds.), *Theoretical foundations of learning environments*, Mahwah, NJ: Lawrence Erlbaum.
- Wimba Inc. (2010). *Wimba Classroom*, Retrieved October 29, 2010, from http://www.wimba.com/products/wimba_classroom/
- Wolf, T. V., Rode, J. A., Sussman, J., & Kellogg, W. A. (2006). Dispelling design as the “black art” of CHI. *Proceedings of Human Interaction 2006, Montréal, Québec, Canada* (pp. 521-530), New York: ACM Press.

Copyright of Journal of Educational Technology & Society is the property of International Forum of Educational Technology & Society (IFETS) and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.