A Design Approach for the Engineering and Construction of CSCL Environments

Kurt Rowley

Advanced Training Technologies Branch
U.S. Air Force Research Laboratory (AL/HRCT)
7909 Lindbergh Dr., Brooks AFB, TX 78235-5352, USA
E-mail: RowleyK@cobra.brooks.af.mil

Abstract

In order to facilitate the design of effective CSCL (Computer Supported Collaborative Learning) environments the design process should include consideration of design factors unique to CSCL such as: 1) understanding the targeted cognitive or performance capabilities in the collaborative context, 2) the technological capabilities of computer supported environments for the specific domain, and 3) the nature and challenges of collaboration strategies that have proven to be effective for improving skills in the relevant domains. A systematic design and development approach that is based on these factors should facilitate the processes of engineering and constructing effective CSCL environments that use advanced information technologies such as AI or ITS. This paper outlines the three CSCL design factors, discusses a systematic design approach that incorporates these factors into a procedure for engineering and constructing effective CSCL environments, proposes a test of the design approach, and raises questions for further discussion related to the use of a design approach to incorporate AI or ITS capabilities into CSCL environments.

Keywords computer supported collaborative learning (CSCL); instructional software engineering;instructional system design models

1. Introduction

Discussion and research related to computer-supported collaborative learning is a relatively new direction in a long tradition of research into collaborative and cooperative learning. Several factors have been identified as important for successful cooperative learning, including heterogeneous teaming, team-based reward structures, rotating roles of team members, individual assessment, and peer tutoring (Johnson & Johnson, 1975; Slavin, 1983). Although the variables of cooperative learning have been studied extensively, it is safe to say that we are still in the beginning stages of research with regard to the design and development of CSCL environments. There is a gradually emerging research base that suggests some heuristics and strategies for designing reliable and effective CSCL environments such as the importance of students being able to contribute comments on each other's work, providing support for peer tutoring or mentoring, and supporting heterogeneous group tasks (Koschmann, 1996; Silverman, 1995; Tomlinson, 1995; Scardamalia, Bereiter, McLean, Swallow & Woodruff, 1989). As the field matures, attributes of successful CSCL environments become known, and both users and developers design many new CSCL environments, it will become increasingly important to learn how to engineer effective CSCL environments.

Some tasks associated with engineering effective instructional systems have been identified, and design models have been adapted to various settings over the past twenty-five years. Some of the early work in developing design models for instruction was focused on large-scale project management using a staged design and development process, with much of the initial work performed for the U.S. Armed Forces. For example, an early Interservice ISD model identified the instructional design task as a step-wise process proceeding through analysis, design, development, implementation, and control (evaluation) stages (Branson, Rayner, Cox, Furman, King & Hannum, 1975). Other models of the design process have viewed the design of an effective delivery system as a classification problem to be solved, and focused on using a systematic, scientific process to classify the skills to be taught in order to help organize methods appropriate to teach those skills (Gagne, Briggs & Wager, 1992). More recent efforts have expanded the views of instructional design to include differentiation between novice and expert design approaches, as well as scaling the design and engineering task to the needs of the design situation. Although the models of the design processes used by novice designers vary in the details, most of them share a common ends-before-means, or backwards-chaining approach to the design process. The simple ADDIE model (Analyze, Design, Develop, Implement, Evaluate) is perhaps the most generic form of these approaches to design, presenting a 'waterfall' style process that starts with the definition of terminal objectives, and works backwards from the objectives through a process of analyzing the learner and environment, designing and sequencing instruction so it will lead to the achievement of the objectives, then developing, formatively evaluating, revising, implementing, and summatively evaluating the instruction. An expert approach typically includes the same design tasks as the novice-level design models, with the exceptions that the expert design process relies more on expert judgment, specific design activities are often more open-ended, and there is usually more use of rapid prototyping and concurrent tasking (Tripp & Bichelmeyer, 1990). The factors that are considered by these design approaches strongly influence the eventual nature of the designed system, and its operation.

In order to develop a model or approach to designing CSCL environments that follows a systematic process it is important to begin with consideration of several unique factors of the CSCL environment, and the elements of typical collaborative instructional strategies that should be considered during the design process. This paper will develop the notion that a design approach for CSCL should incorporate an instructional engineering process consistent with traditional instructional design models. The particular design model developed will be based on a standard ADDIE design model, with the addition of design factors related to the construction of effective CSCL environments. A test of the design approach will be proposed, and a few questions related to using the design approach to engineer and construct a CSCL environment will be raised.

2. CSCL Design Factors

Collaborative learning environments bring with them many natural challenges. For example, rather than a traditional single channel of communication between teacher and student, collaborative environments often require extensive interactions between students, and between groups of students and instructors, significantly changing the structure of the learning environment. Some skills may need to be supported differently than others in a collaborative environment. For example, supporting the skill of painting in a collaborative environment may work best with an evaluative collaboration, whereas supporting the development of a mathematical skills may work best with a peer tutoring environment. Designers of CSCL environments must attend to many issues related to these types of challenges associated with collaborative learning environments.

Some of the specific factors to consider during the design of a CSCL are related to potential interactions between content of the subject (the domain), and the treatment (collaborative instructional strategy). This means that the nature of the skill to be learned should be understood in the context of computer supported collaboration. It is also important to understand the level of support required from the technological components of the CSCL. Finally, many collaborative strategies have been tested in various domains, and some have proven to be more effective than others. The following are some descriptions and examples of factors important in the design process for engineering and constructing a CSCL including defining the skill in a collaborative context, understanding the technological capabilities, and discovering proven collaborative strategies for a particular domain.

2.1. Skill Specification in a Collaborative Context

What are the specific skills to be learned by the students for the CSCL and what are some of the possible collaborative interactions for learning those skills? Processes for the analysis of skills have been well identified in terms of traditional instructional design approaches to task analysis, and more recently, with cognitive task analysis. The skills needed must be understood in terms of possible collaborative strategies for teaching the skills, in order for the end goal of the design of the CSCL to be achieved. Defining the skills in a collaborative context will also help facilitate a design process for the CSCL, and will lead to an environment that is centered around the accomplishment of the instructional objectives of the CSCL environment. Once the desired changes in cognitive processes and performance capabilities are stated, and collaboration options for the learning and practice of the skills are understood, then a newly designed CSCL environment will more likely support these changes.

As an example, consider the possible collaborative strategies for a CSCL for teaching English composition skills. The USAF *MAESTRO* Writing Process Tutor teaches skills needed to improve writing abilities (Rowley & Crevoisier, 1997). A specific goal for those improved abilities has been identified through protocol analysis and extensive cognitive research. The goal in this case is the development of a procedural skill used by experts in order to set writing goals, draw on memory and outside sources for writing plans and domain information, generate ideas, organize those ideas and translating them into draft text, then iteratively review and revise the text while considering the writing goals (Flower & Hayes, 1980). The collaborative possibilities for learning and practicing these skills are significant at every stage of the writing process. Peers can collaborate on joint documents as a team, for example, with each team member performing one stage of the writing process. Peers may also provide peer tutoring, depending on their level of literacy and expertise in the writing process. In order to support this type of collaboration, we would first determine the desired improvements in writing skills, and then develop collaborative strategies and CSCL design around the accomplishment of that performance improvement.

2.2. Technological Capabilities for Domain

The ability of computer technology to provide collaborative support varies with the types of activities to be performed. There would be special technical considerations for supporting collaborative learning while teaching sports in a team environment, for example. Field-based support for sports might include use of remote data display devices for coaches or team leaders, using wireless network connections. On the other hand, there may be team activities that can be practiced in a cooperative environment away from the playing field in a computer lab game simulation. Common technological capabilities available for CSCL environments include synchronous chatting, asynchronous messaging, supporting group tasks, project and work group coordination, peer review of work with comments, use of multiple-person interfaces for complex simulations, and so forth. Each of these technical capabilities comes with its own unique challenges. Some of these challenges have been studied and addressed directly, such as a human factors study showing that users in collaborative chatting environments prefer tiled multiple-window designs when cognitive requirements are high, and single windows when cognitive requirements are low (Ahern, 1995). There are also design cases that provide models for structuring the CSCL technologies, providing experience-based heuristics for designing CSCL environments and suggesting technologies to support them. Several such heuristics in an adult learning context include the importance of providing an open environment in which students have many choices over content, allowing self-management, help students pursue authentic purposes, supporting extensive student interaction, and facilitating collaborative assessment (McConnell, 1995). Specific technological capabilities of a CSCL vary tremendously, so before designing the CSCL it is important to assess the capabilities required.

In order to identify the technologies that can be used with a CSCL the technical requirements should be identified. Technical requirements could include: 1) computing requirements for the expected implementation system, 2) pedagogical requirements of the selected collaborative learning strategy; and 3) connectivity or networking requirements for student interactions. For example, when converting *MAESTRO* to a CSCL, the expected computing requirements may include the use of the Internet, with the expectation that the delivery environment will be a web browser, with a server for providing an inference engine. The pedagogical requirements will determine the level of sophistication of the CSCL implementation. For example, in a classroom-based CSCL, this may include factors such as the amount of time that the teachers can allocate to allow the students to use the CSCL environment during class time, or at home. Pedagogical requirements will also include the amount of support or coaching needed during collaboration. Finally, once the computer requirements and pedagogical requirements are identified, the connectivity requirements can be mapped out based on expected communication patterns. This will help ensure that the necessary interaction capabilities for the pedagogical considerations are designed into the CSCL environment. Representation of the connectivity requirements might include, for example, generating diagrams that can guide the engineering and construction of the CSCL environment.

2.3. Proven Collaborative Strategies for a Domain

The nature and challenges of collaborative learning environments have been well documented in work by Slavin (1990), and Johnson & Johnson, including a meta-analysis of 122 studies in cooperative learning strategies (Johnson, Maruyama, Johnson, Nelson & Skon 1981). The general finding of the meta-analysis was that cooperative or collaborative group structures are more effective than individual and competitive group structures, regardless of how the comparative individual or competitive group situations are structured, and regardless of academic subject area of the studies (the meta-analysis included language arts, reading, math, science, social studies, psychology, and physical education). The superiority of cooperative grouping is not universal, however, and many design factors influence the success of a particular effort. For example, Johnson et. al. found that cooperative group learning structures had negligible effects on rote learning or memorization tasks. Their multivariate statistical analysis also only explained 21% of the variance, leaving a significant number of mediating and moderating variables unidentified. Some of the unidentified variables may include challenges commonly identified by practitioners and researchers in cooperative learning environments. Based on the author's experience with cooperative groups, as well as discussions with other teachers who have used cooperative groupings extensively, some of these challenges may include tension in the social context of the group, variation in the efforts of team members, variability in the ability of the teacher to instigate social bonding within the groups, additional facility requirements for the group's meetings, a high negative impact of absenteeism on groups, extra workload on the teacher due to the need to teach group process skills and regulate variations in group performance, and increased curriculum planning requirements.

However, in spite of these challenges, cooperative and collaborative learning strategies have proven successful in many domains. For example, there is a significant body of research comparing instructional strategies for teaching written composition. In a meta-analysis of 60 studies comparing instructional approaches, Hillock (1986) compared cooperative, problem-based learning environments with traditional (lecture-test), naturalistic (like constructivism, discovery oriented), and individualized (programmed instruction) environments for teaching English composition. He found that cooperative, group-based instructional treatments far outperformed other environments, with a normalized main effect size of .44 of a standard deviation across the studies. By comparison, the alternative approaches produced effect sizes of .02 (traditional), .19 (natural discovery), and .17 (individualized). What were the factors identified in these successful cooperative and collaborative learning environments? Some of the key design factors for the collaborative environments Hillock studied included "(1) the use of clear and specific objectives, e.g., to increase the use of specific detail and figurative language; (2) materials and problems selected to engage students with each other in specifiable processes important to some particular aspect of writing; and (3) activities, such as small-group problem-centered discussions, conducive to high levels of peer interaction concerning specific tasks." (Hillock, p.122). He found that the teachers using these cooperative and collaborative strategies typically provided brief introductory presentations, then had students work in small groups on small tasks, and often had students continue working independently on follow-up tasks after the group work was completed. In essence, these approaches engaged the students in concrete problems, with peer feedback during the critical first application of new skills, and then allowed the students to continue improving the new skills on their own, preparing them for the eventual autonomous performance of their new writing skills.

So what can we learn from this example? In order to design an effective CSCL environment it is useful to consider the large amount of research into effective collaboration strategies for specific domains. This can both help validate the effectiveness of a collaborative environment, and outline where and how best to apply technology in the support of a collaborative learning environment.

3. CSCL Environment Design Model

A systematic design model for designing a CSCL environment should take into consideration the CSCL design factors outlined above along with traditional design concerns. If we wish to incorporate advanced technologies such as AI, it will be important to also consider the capabilities of those technologies within the CSCL framework.

3.1. Designing for AI in CSCL Environments

Among the many applications of AI techniques that can be used in support of computer-assisted instructional systems, expert systems may be particularly useful for CSCL decision-making. Some instructional decisions to be made for a CSCL may relate to when and how to assign collaboration, what level of collaboration to seek, how to control the instructional strategy, and possibly how to utilize collaboration to provide student assessment and feedback to the tutor. Automating these decisions through advanced AI technology could be particularly useful in complex domains such as writing where there are limits to the ability of the instructional software to reduce student outputs to computational form. In these types of collaborative environments the system can make inferences based on the processes that the student follows, rather than student products. The system can also make inferences based on questions the student might ask, the number and types of words used in a given workspace, or time spent in an area of the CSCL. For example, following a simple analysis of the length of a student text, an expert system might recognize that the student has spent too long revising the document given the size of the document, and determine that a student could benefit from peer review of a document, or peer tutoring.

In order for a CSCL designer to apply and benefit from the capabilities of AI such as expert systems, the CSCL design approach should lead to a consideration of the collaborative contexts within which AI capabilities might be useful. These contexts could be considered during the initial Analysis and Design stages of a standard ADDIE design approach. Additionally, the use of AI to help select specific collaborative strategies should be considered during the Design and Development stages of the design approach. Other capabilities of AI should also be considered during the planning stages of implementation as they might suggest useful and novel means of implementing the collaborative strategy. For example the use of an AI-style autonomous agent in a CSCL environment might be very helpful in some domains, or in training situations in which the student must work alone, or access to collaborators is restricted. The agent could 'fill in' for the missing participant.

3.2. Systematic Approach to Design of a CSCL

Models for designing instruction tend to be generalizeable as the ADDIE model (Analyze, Design, Develop, Implement, Evaluate), which is a multi-step process in which design flows from a situational analysis and identification of objectives. In essence, it is a backwards-chaining process, starting with the definition of the final instructional goals, identifying test items for those goals, then designing and developing instruction that sequences perquisite materials appropriately to lead the student to successful skill development and accomplishment of the end goals. Finally, the ADDIE model addresses delivery system implementation and summative evaluation. Most ADDIE-type design models include formative evaluation of some sort after each step of the process. This is a systematic process for design, but it is not necessarily focused on construction of any particular type of delivery system. An ADDIE approach is quite open-ended in terms of the nature of the designed environment. Figure 1 illustrates an ADDIE-type design model that includes both traditional design tasks and the CSCL design factors identified above (underlined in the Fig. 1 model).

By following this type of design approach, a CSCL environment can be constructed in a fashion that deliberately addresses CSCL design factors. The ADDIE process provides a useful organizing concept for novice designers and it can provide continuity during the design of CSCL environments, ensuring that all aspects of instruction contribute toward the accomplishment of the instructional goals, or 'terminal objective'.

Figure 1: Adapted ADDIE Design Approach

4. Testing the CSCL Design Approach

Although this CSCL design approach is a simplified representation of the design process, it illustrates the primary characteristics of designing a CSCL environment. An ongoing study in which an existing LAN-based intelligent tutor is being re-engineered into a collaborative, Internet-based distributed learning environment will provide a test of the design approach for engineering and construction CSCL environments (Rowley, 1997). The CSCL design approach is being tested in the context of the revision of *MAESTRO The Writing Process Tutor*. *MAESTRO* is part of a U.S. Air Force technology transfer project. The original MAESTRO Writing Process Tutor research was conducted through the U.S. Air Force's Fundamental Skills Training (FST) Program. FST is a multi-year research effort to transfer advanced, adaptive training technology capabilities developed under Air Force technical training research to public education. *MAESTRO* teaches the procedural skills of an expert writer, based on cognitive research into the writing process, and several years of evaluative teacher feedback. MAESTRO was written in Asymetrix Toolbook and implemented successfully over LANs in computer labs at 12 High Schools in the U.S. during the 1996-97 school year in a traditional controlled experiment. The tutor was implemented as a regular part of the writing curriculum. The tutor was designed to be integrated with the classroom. *MAESTRO* supports writing assignments selected or supplied by the English teacher. Ongoing research is addressing the effectiveness of *MAESTRO* (Rowley & Crevoisier, 1997).

The new *MAESTRO* tutor will include the capability of peer interaction during the student tutoring sessions, as represented by the 'Peer or Team Member' box on the far right of the diagram in Fig.2. This will include the capability for peer revision as part of the writing process, which is a well-documented collaborative instructional strategy. Additional collaborative strategies will be facilitated by the addition of features such as document sharing, synchronous chatting, and group editing of documents. The new version of the tutor will use a WWW-based Interoperable distributed learning environment (DLE) architecture (Rowley, 1997). The CSCL design approach will be followed during the design of instructional resources, designing knowledge bases, and developing generic rules for the new version of*MAESTRO*.

Figure 2: Interoperable DLE Architecture

5. CSCL Design Approach Questions

There are many theories related to the role of the social context in individual and classroom learning, cooperative learning, and teaming, but there is currently little in the way of prescriptive theory for the design of CSCL environments. Further research into the events of successful collaborative learning would be useful in order to provide prescriptive theory. An empirically-based collaborative instruction theory would be useful for further development of this CSCL environment design approach.

Additional questions for further discussion and research are related to skills that the developer must possess in order to engineer this type of environment successfully. What unique skills must the developer acquire to use the ADDIE process to design a CSCL environment? Also, how can the developer best accomplish knowledge engineering for the application of AI to the design of a CSCL environment in order to accommodate some of the unique traits of the CSCL learning environment?

Finally, questions related to the proximity of collaborators in a distributed implementation of a CSCL environment should be explored further. For example, are there unique design factors that must be taken into consideration when collaborating students are remote from each other, or when students occasionally meet face-to-face while using a CSCL environment?

In order to address some of the ideas presented in this paper, the CSCL environment design approach will be tested, the success of the approach evaluated, and the CSCL design approach will then be revised as indicated. The engineering and construction of CSCL environments must eventually be reduced to a reliable design approach in order for a sufficient number of CSCL environments to be developed to have a long-term effect on education and training. The development of a reliable design approach that accommodates the application of AI to CSCL environments should become important in preparing practitioner-level technologists to successfully produce effective CSCL environments.

REFERENCES

Ahern, Terence C. (1995). Effect of window state on user behavior in an on-line computer mediated conference. Paper presented at CSCL 95, October, 1995, Indiana University, Bloomington.

Branson, R., Rayner, G., Cox, J., Furman, J., King, F., and Hannum, W. (1975). *Interservice procedures for instructional systems development* (5 Vols.) (TRADOC Pam 350-30 and NAVEDTRA 106A). Ft. Monroe, VA: U.S. Army Training and Doctrine Command.

Flower, L., and Hayes, J. R. (1980). The dynamics of composing: Making plans and juggling constraints. In L. Gregg, & E. Steinberg (Eds.), *Cognitive processes in writing* (31-50). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gagne, R., Briggs, L. & Wager, W. (1992). *Principles of instructional design*. Fort Worth, TX: Harcourt Brace Jovanovich College

Publishers.

Hillock, George. (1986). *Research on written composition*. Urbana, IL: ERIC Clearinghouse on Reading and Communication Skills.

Johnson, D. & Johnson, R. (1975). *Learning together and alone: Cooperation, competition, and individualization*. Englewood Cliffs, NJ: Prentice-Hall.

Johnson, D., Maruyama, G., Johnson, R., Nelson, D. & Skon. (1981). Effects of cooperative, competitive, and individualistic goal structures on achievement: A meta-analysis. *Psychological bulletin, 89*(1), 47-62.

Koschmann, Timothy, (Ed.). (1996). *CSCL: Theory and practice of an emerging paradigm.* Mahway, NJ: Lawrence Erlbaum Associates, Inc.

McConnell, David. (1995). A methodology for designing post graduate professional development distant learning CSCL programmes. Paper presented at CSCL 95, October, 1995, Indiana University, Bloomington.

Rowley, Kurt. (1997). A generic instructional vocabulary for an interoperable distributed learning environment: Adapting *MAESTRO* the Writing Process Tutor. *AI-ED 97 Workshop: Issues in achieving cost-effective and reusable intelligent learning environments, Kobe, Japan, August, 1997.*

Rowley, K. & Crevoisier, M. (1997). MAESTRO: Guiding students to skillful performance of the writing process. *Proceedings of ED-Media 97, Calgary, June, 1997*.

Scardemalia, M., Bereiter, C., McLean, R., Swallow, J., and Woodruff, E. (1989). Computer-supported intentional learning environments. *Journal of educational computing research, 5*, 51-68.

Slavin, R.E. (1983). *Cooperative learning*. New York: Longman.

Slavin, R.E. (1990). Research on cooperative learning: Consensus and controversy. *Educational Leadership*, *47* (December 1989/January 1990), 52- 54.

Silverman, Barry. (1995). Computer supported collaborative learning (CSCL). *Computers & Education, 25*(3), 81-91.

Tomlinson, Helen. (1995). Computer supported collaborative learning in schools: A distributed approach. *British Journal of Educational Technology, 26*(2), 131-140.

Tripp, S. & Bichelmeyer, B. (1990). Rapid prototyping: An alternative instructional design strategy. *Educational technology research and development, 38*(1), 31-44.