Epistemic Forms and Epistemic Games

- **Full citation:**

  Sherry, L., & Trigg, M. (1996). Epistemic forms and epistemic games. *Educational Technology, 36*(3), 38-44.

Lorraine Sherry (lsherry@carbon.cudenver.edu)
Maggie Trigg (sis_mtrigg@Chrisy.arapahoe.edu)

Abstract

Though critical thinking is given much lip service by teachers and instructional designers, the process of facilitating it is not always transparent. This paper is intended to help teachers and instructional designers encourage reflection, analytical thinking, and inquiry learning in students, by playing epistemic games. These are reflective, knowledge-generating activities that provide learners with new ways of looking at facts, concepts, and events, as well as clarifying the relationships within the new material. Epistemic games are applicable to the knowledge-building process of instructional design and software development, as well as to the classroom. Several examples are provided, from a variety of subject areas. A method is given for educators to organize material into appropriate epistemic forms, and to design and create their own epistemic games.

What are epistemic games?

Epistemic games are reflective, knowledge-generating activities that can be used in any classroom, and can be created for any knowledge domain. The theory of epistemic games was developed by Collins and Ferguson (1993) and Morrison and Collins (1995). It is a formalized procedure for constructing knowledge representations, in the spirit of Merrill's Component Display Theory and Riegeluth's Elaboration Theory, where the type of structure used to organize and integrate new knowledge is related to the desired learning outcomes (see Saettler, 1990, p. 348-9).

Though instructional theories are important in and of themselves, they must mesh with classroom practice to be of practical value. Our purpose here is not to repeat the excellent theoretical treatment presented by Collins and Ferguson (1993) and Morrison and Collins (1995), but rather, to bring theory closer to practice by providing teachers and instructional designers with a conceptual framework and a method to do what they have informally been doing all along.

Instructional designers realize that not all students need explicit support for knowledge construction. However, many students struggle with abstract concepts, and also with the intermediate steps in problem-solving which the textbook leaves out, considering them to be "intuitively obvious to the casual observer". Epistemic game-playing enables students to create explicit epistemic forms Ñ i.e., knowledge structures such as matrices, block diagrams, decision trees, etc.Ñ which they can use to make distinctions, tabulate facts and concepts, and then attempt to make sense out of them.

Once students begin the process of building knowledge structures that visually illustrate the rules of the domain, they start to consider education as fun and enjoyable rather than a mindless chore, and learning inevitably follows.

Some Definitions

Let's start by defining some key terms.

An *epistemic form* is a target structure that guides the inquiry process. It shows how knowledge is organized or concepts are classified, as well as illustrating the relationships among the different facts and concepts being learned. The completion or creation of the structure is the object of the epistemic game.

An *epistemic game* is a set of moves, entry conditions, constraints, and strategies that guide the building of the epistemic form. The rules may be complex or simple, implicit or explicit.

Epistemic fluency is the ability to identify and use different ways of knowing, including different epistemic forms and various epistemic games. It is the ability to organize information into multiple patterns, to participate in multiple ways of making sense of the world.

Some Sample Epistemic Forms

Models of information, or epistemic forms, are already used in classrooms, and are familiar to all of us. A simple epistemic form is a family tree. It is a way of visually organizing family information into a meaningful and recognizable structure, one that shows the relationships among family members.

We use epistemic forms every day. Businesses use organization charts to show the chain of command within their corporation. Maps show how geographic areas fit together. Calendars are used to impose the structure of days, weeks, and months, on time. Recipes are organized into two areas: an ingredient list, and a process described by a series of instructions. Systemic training design models are used by instructional designers to show how the various facets of the ISD process relate to one another.

Epistemic forms are used in all subjects taught in high schools and colleges. Chemists organize elements into the periodic table by atomic weight. Historians depict events by placing them on a timeline. Mathematicians show the relationships between different variables by constructing equations. Economists show the relationship between supply and demand by drawing a graph. Musicians arrange musical notes, and specify their loudness and duration, to create musical compositions. Computer scientists and software developers use animated simulators to visualize data structures, algorithms, and computer architecture, and to trace the flow of data through a computer.

Characteristics of Epistemic Games

An epistemic game is the process of completing an epistemic form or knowledge structure. Instead of being given the completed structure in a lecture or a handout, students create the form by filling in the slots of the structure themselves.

Entry conditions determine when the epistemic game is appropriate. They include the assumed knowledge level of the students who will be playing the game. They also provide a means of determining whether or not the

Moves are actions that can be taken at different points in the game They can include moving a piece of a diagram from one location to another, decreasing the value of a variable in a system, or adding an item to a list. Moves can be explicitly spelled out, or they can be determined by the students while playing the game.

Rules or constraints focus and facilitate usage of the epistemic game, by determining how to fill the slots in the target form. They determine such qualities as completeness or uniqueness of items, how and when items are related, and how the parts of a system affect each other. Rules determine if a move toward building the knowledge structure is legal or useful.

Transfers are changes to other epistemic games. They occur when the original game is no longer appropriate for the information, and a new target structure is required.

A Typical Epistemic Game: the List Game

The easiest game to deconstruct is the list game. It is played by asking a question and listing the answers. If the question requires answers that need to be discovered, rather than looked up, then making the list is an act of creating new knowledge.

The epistemic form is a list that answers the question.

Entry conditions are prerequisite skills or knowledge which students must have prior to playing the game, i.e., the facts or events upon which the question is based.

Moves or actions that can be taken during the list-making process are: add a new item, combine two or more items into one, change an item, split an item into two or more items, or delete an item.

Rules or constraints that govern the list game are: similarity (items must have equal size, importance or other critical characteristics), coverage (the list must include all possible answers), distinctness (each item on the list is unique), multiplicity (the question has more than one answer), and brevity (the list must be short enough to be comprehensible).

A list can be transformed into many other forms: a matrix, a hierarchical chart, a time line, a spreadsheet, or whatever else becomes appropriate. The learning group can transfer to a new form as required.

Categories of Epistemic Games

Collins and Ferguson (1993) categorize epistemic games into three types: structural analysis games, functional analysis games, and process analysis games. These present increasing levels of challenge to participants Ñ structural analysis games being easiest, and process analysis games being the most difficult.

Structural analysis games determine the components or elements of a system. Examples include making a list,

Functional analysis games show how the elements in a system are related to each other. Examples include: creating a hierarchical chart, deriving an equation, diagramming a sentence, or making a causal-chain diagram.

Process analysis games describe how a system behaves. Examples include: drawing a program flowchart, determining a system of equations, graphing the change in a system over time, or creating a spreadsheet to project business profits.

These three categories closely parallel the three ways of structuring knowledge as described by Reigeluth's elaboration theory: conceptually, procedurally, and theoretically (Reigeluth, 1983; 1987). A conceptual knowledge structure shows us what something is; a procedural knowledge structure shows us how something works; and a theoretical knowledge structure shows us why something works. These structures draw on declarative, procedural, and structural knowledge, respectively.

Another way of grouping epistemic forms is by their physical attributes, i.e., graphs, tables, time charts, maps, structure diagrams, process diagrams, and network charts (Lohse, Biolsi, Walker, & Rueter, 1994). Still another way of grouping has more to do with the type of information that is being represented: spatial or non-spatial; temporal or non-temporal; concrete or abstract; continuous or discrete; numeric or non-numeric; and static structure or dynamic process.

Epistemic games can combine more than one type of structure. For example, MPC1, a computer-based epistemic game for depicting the role, function, structure, and mechanism of a model computer, is a form-and-function analysis game (Sherry, 1995). A map of Europe representing the dominant powers in 1918 or 1944 can be considered a structural analysis form as well as a functional analysis form. The Fourth Generation ISD model integrates structure with process to describe the dynamic relationship between five authoring activity domains: analysis, design, production, implementation, and maintenance (Seels, 1995, p. 122).

Examples of Epistemic Forms

Epistemic forms can range from formal structures that are developed only for specific disciplines, to generalized and informal arrangements used in many different disciplines. These include, among others:

- list
- matrix or table
- molecular structure model
- DNA model
- periodic table of the elements
- database
- graph or chart
- nitrogen cycle diagram
- pert chart
- GANTT chart
- simple binary tree
- floor plan
- organizational chart
- musical score
- trend graph
- iconic menu for a software package
- instructional design model
- cognitive map
- timeline
- cause/effect diagram
- problem/solution diagram
- map
- network
- stage model
- cost/benefit diagram
- star diagram
- sentence diagram
- outline for a paper or report
- a menu page in a hypertext document

and many, many more.

Two examples of epistemic games for the classroom

1. The hierarchy game

The hierarchy game is a functional analysis game, based on a hierarchical or tree structure. Entities must be distinct and unique. Information must be able to be organized in multiple levels, forming subordinate, superordinate, and possibly coordinate relationships. Legal moves are: add, change, or remove an entity; move an entity from one level to another; add, change, or remove a level; move a level to another location.

Arranging food products into family groups is appropriate for players who have seen traditional food charts, and who are familiar with the vocabulary used. Index cards containing terms such as meat, dairy, legumes, plants, nuts, eggs, leafy, animals, fruits, etc., are stacked randomly. They can be placed on a posterboard with temporary adhesive.

Players are grouped into teams of three. Each player draws one card per turn, and places it onto a hierarchy diagram on the posterboard. The player then draws connecting lines to join the new card to existing cards.

For example, the first player draws the card "meat" and places it on the posterboard. The second player draws the card "plants" and places it on the same row with "meat". The third player draws the card "animals" and places it on the same row with "meat" and "plants". The fourth player moves "meat" to the row below. Subsequent players draw cards and place them on the posterboard according to the legal moves.

This can be used for any topic that involves classification, such as rocks and gems, students' town government, motor vehicles, or arranging files and subdirectories in a computer literacy class.

2. The temporal decomposition game

The temporal decomposition game is a process analysis game, based on a stage model diagram. It can be played whenever the stages of an event or process are separate and unique. Players should have read about the event or process being examined. Legal moves are: add, change, or remove a stage in the event; add, change, or remove a character in or a component of the event; add, change, or remove a connection between the event and the character or component; combine two stages; or split a stage in two.

The rules and constraints are: the connection between the stage and the character of component must be specified before play begins; no stages may overlap; all stages must be included; players will determine the time intervals between stages; stages must be chronologically arranged; and characters and components need not appear in every stage.

The temporal decomposition game can show how the plot of Snow White unfolds from the perspective of the main characters in the story. Players must have read Snow White, and be able to operate spreadsheet software or create tables using word processing software.

Players are divided into groups of three or four, each with a computer and either spreadsheet or word processing software. They can make any of the legal moves (above), unless other moves are added by consensus of the group. The order of play can be determined by any random process such as rolling dice or spinning a needle. The connection between the characters and the plot events are the emotions felt by each character. The appropriate emotion will be entered into the cell for each character, in each stage of the plot.

For example, the first player adds "Snow White" as a character. The second player fills in the first event in the story, "Snow White being sent away by the wicked Queen". The third player adds the character "Queen". The next player adds the event, "the hunter returns to the Queen". The next player changes that event to "Snow White finds the dwarves' house". The players talk about the level of plot detail, and decide that only events involving two or more main characters should be included. Subsequent players begin adding the feelings of each character at each stage.

A variation on this game is the examination of the major causes and events of World War II. List the years chronologically across the column headings, and the countries down the row headings. The connections are the

Other examples of epistemic games

Here are just a few of the many possibilities for games that can be played at many levels, and in many different subject areas:

- creating a system block diagram or a flowchart
- designing a floor plan
- creating a theory by presenting evidence for and against
- drawing a problem/solution diagram (antecedent events help identify possible solutions)
- developing multiple perspectives by role playing
- deriving rules
- creating a cross products table for comparisons across dimensions
- combining elements to form larger units
- drawing a core diagram with concentric circles representing importance or priority, with layers radiating out
- pairing games: opposites, synonyms, analogous terms, translations
- setting up truth tables
- writing an outline
- developing a concept map
- building a regression analysis model
- determining functions necessary to update a database
- debugging a computer program using a source-level debugger
- creating a network diagram with nodes and links.

Putting it all together: a case study

As a typical scenario, let's consider that you are teaching a middle school science class on ecology. Looking through the textbook, you write down the important terms in the chapters your students have read, e.g., water, ice age, rain forest, grass, rocks, car exhaust fumes, animals, birds, mountains, blizzard, reservoirs, earthquakes, wilderness areas, trees, extinction, mammals, and so forth. A network diagram with circles containing single concepts, called nodes, which are connected by lines, would be ideal. Each line describes one of six different types of connection: characteristic, leads to, type of, analogy, evidence, or part of. Since "analogy" doesn't seem to fit your schema, you replace "analogy" with "affects".

You assign a specific color for to type of line, gather six markers and a flip chart, locate a dice to roll for turn-taking, and then divide the players into groups of three. Players look through the word list to find one that can be related to the existing network, using one of the six connections specified.

When the groups have completed their network diagrams, they will have come away from the experience with a wonderful aid for writing papers or studying for exams.

Designing Epistemic Games

Instructional designers must be creative to design good epistemic games. With a bit of insight, it's possible to find an appropriate epistemic game for just about every subject. Here's a good way to begin.

Start by looking at the content to be learned. Is there a structure that is already commonly used to represent the material? If not, are there any diagrams or content organizers in the text, instructor's guide, or ancillary materials?

Look for an existing structure that appears to fit the content. If one doesn't exist, you can use a frame game to create one. Frame games are described in Jonassen, Beissner, & Yacci (1993). Frames are geometric shapes that are connected by lines or arrows, if necessary, and which represent target structures, namely, hierarchy, sequence, part/whole, association, analogy, equality, pair of opposites, similarity, and cause/effect. Then list the important terms and concepts that you would like to use in your epistemic game. Fill in the blank frames with some sample items, and look for the best match between content and form.

Draw the empty form and start filling it in. Pay careful attention to the manner in which you put your data into the slots of the blank form: adding, changing, removing, and rearranging it. This will determine the rules and constraints for the game.

Analyze what prerequisite information and skills are required to fill in the target form. This will help you specify the entry conditions.

Consider what media you will use to depict the external knowledge representation, and how you will determine turn-taking.

Decide whether you will provide the raw information to the players, or whether they will need to find it themselves.

Evaluate your game. Try playing it yourself first, and then make improvements as needed.

After the game

After the exercise is finished, it is important to explore what the students actually learned by playing the game. Was the chosen form the best representation of the given information? Why or why not? What might be better? Were modifications necessary as the game progressed? Review the explicit relationships discovered, and ask students to give examples of various concepts.

Investigate the cognitive side of game playing as well. Did it help students see relationships between concepts in the material? Were they able to identify patterns in the material? Can anyone think of other information that could be represented by this form? Then follow up the reflective process by active experimentation, by giving students some problems to solve.

Finally, take time to stop and think about what happened before, during, and after the game. Review the entire process to evaluate the strengths and weaknesses of the game. How can you assess whether the material was learned? Did the activity help promote a high quality learning environment? And, overall, how successful was the activity? Answering these questions will enable you to determine how useful the game actually was, within the context of the students' learning environment and the prior knowledge they brought into the game-playing situation, and to revise it accordingly.

Visual Programming: Epistemic Games and Design

Epistemic games and forms are not limited to the classroom; there are many applications in the workplace as well. Because these games have formal rules, moves, and structure, computer-based technology can often be used as a tool or vehicle to represent them.

Computer scientists use visual programming environments, especially animated structures, to display the flow of abstract data through the computer and its assorted peripheral devices; to step through algorithms such as sorting, searching, and hashing routines; to manipulate graphics; and to develop a visual conception of computer processing. This is especially important in software development, where program construction requires abstraction, analysis, and synthesis on the part of the designer. The same rules apply here as they do to the classroom Ñ look for relationships; identify patterns; see if your representation is the best one for solving your problem or depicting your knowledge structure.

Software engineers and designers are beginning to find that one animated figure is worth many static pictures, and are currently developing some extremely sophisticated process analysis games. Mayerhofer and deLucena (1992) have devised an Algorithm Simulation and Animation Environment (ASA) in which students write Pascal programs using classic algorithms and data structures, and then run them stepwise. The results are shown on a screen as a graphic that changes its structure with each program step. Petzold (1990) has used a Graphics Programming Interface (GPI) to allow software developers to experiment with different coordinate transformation formulas for manipulating graphic metafile images. This enables users to try out different mathematical formulas Ñ different moves Ñ to modify the coordinate points of the target graphic. After each new transformation, the software displays the resulting image. Zernik, Snir, and Malki's prototype visualization tool provides programmers with an overall view of parallel program execution. By "freezing time", software designers can sidestep from one processor to another, and run the program forwards or backwards, step by step, even for systems with thousands of processors. In each of these examples, every new transformation formula or operation represents another step in the game; it operates on the target form to change it in some way.

Johnson (1995) has used a PC-based dynamic simulation, titled MPC1, to enable students to make the transition from a mental model of a computer to the external reality of data processing. MPC1 has also been used in a graduate Instructional Technology hardware course. It helps novice instructional designers and corporate trainers comprehend how authoring programs deal with answer-judging or process the script that they write, by introducing them to the basic operation of a simplified computer. As designer-trainees work with the simulation, they begin to comprehend the role, function, structure, and mechanism of a real computer:

- role (the part played by each component, such as RAM, registers, and I/O buffers),
- function (the goal that each component accomplishes),
- structure (the computer architecture), and
- mechanism (the process by which the structure accomplishes the function).

MPC1 works just like a programmable calculator that is limited to integer arithmetic. However, it contains many essential components of a typical microcomputer: I/O buffers, RAM cells, three registers Ñ accumulator, instruction register, and program counter Ñ and a message status box. There is a set of ten operation codes which are used to write simple machine language programs. These instructions can be thought of as the set of moves in an epistemic game. When the user runs the program, the software displays the movement of data through the model computer as it executes the program.

We'll show you a simple example and explain how it works. Figure 1 depicts a screen capture of the Fetch-Execute Cycle for the fourth of seven instructions in a machine language program that is supposed to double an integer. The user has already written and entered the program, input a data value (here, 276), and executed the first three instructions. The program instructions are located in RAM Cells 0 through 6. The program counter shows that the instruction from RAM Cell 3 has just been fetched to the instruction register, and the program counter has been incremented from 3 to 4. The machine code operation in the instruction register indicates that the model computer will now "store the current contents of the accumulator in RAM Cell 62". The screen display shows the user that the value 552 is now located in both the accumulator and RAM Cell 62.

[insert figure 1 about here]

Figure 1. A screen print from the MPC1 model computer simulation.

The MPC1 software "grows with the learner". As users become more adept with the language, they soon become quite expert at writing sophisticated programs to solve complicated integer arithmetic problems. Moreover, since the software allows them to debug flawed programs by executing them stepwise, they can see the exact point at which they made a mistake in their code. The beauty of this simulation environment is that it can be used by people in many different fields, because it enables the user to form a dynamic mental model of what actually goes on inside a computer. After experimenting with the software for an hour or so, instructional designers report that they have gained a much clearer comprehension of the basic principles which govern how their CBI programs, stacks, or multimedia training applications work.

Why Use Epistemic Games?

Epistemic game playing teaches students how to construct and organize their own knowledge. When students create their own epistemic forms, they are analyzing the material and synthesizing new structures that show the relationships within the material. Seeing how information can be organized into various structures promotes fluency in pattern recognition, a skill that is associated with expert behavior and creativity.

Morrison and Collins (1995) believe that epistemic fluency is essential in our complex, multicultural society. People who are familiar with many ways of viewing things are more likely to communicate clearly across cultural boundaries. Perry, too, considers the ability to see and compare more than one viewpoint as a prerequisite of a mature, relativistic, world view (Perry, 1985).

Game playing also teaches metacognitive skills Ñ awareness of relationships among things, and different ways of looking at things. It enhances the learning experience by helping students visually organize new material, and by making them active participants in the learning process. Games provide a reflective activity that can be done individually or in collaborative groups.

Epistemic game playing can be viewed as critical to those who wish to participate productively in our increasingly complex technological society. Reflective cognition - the restructuring of knowledge or changing schemata - is how new concepts are learned (Jonassen, Beissner, & Yacci, 1993). Expert knowledge depends on how well facts and concepts are arranged into patterns. Concept learning can be very difficult and require much effort, but providing an epistemic game to assist students in concept building can make it easier.

According to Norman (1993), there are three kinds of learning: accretion (accumulating facts), tuning (solving problems), and restructuring (organizing knowledge to understand a new concept). He has coined the term, *cognitive artifact,* to describe any external representation of an abstract concept that help us restructure knowledge. Restructuring is the most difficult part of learning, so any tool that helps with knowledge restructuring is valuable, whether it be in the classroom or in the workplace.

Game playing also makes learning easier because it motivates learners. It helps them take an active role in learning, and in developing their own theories of what knowledge consists of, by allowing them to create the target form themselves. (see Bereiter & Scardamalia, 1989). The dynamic creation of the knowledge structure is far more interesting and engaging to learners than is being asked to memorize the completed structure.

Conclusions

Epistemic forms and games are powerful aids for algorithmic reasoning and for the design process. They also provide a theoretical basis for a great variety of knowledge-structuring activities that can be used in any classroom. Moreover, the same principles generalize to software development, instructional design, and potentially to any academic, corporate, or workplace domain where knowledge construction and representation is involved.

References

Bereiter, C., & Scardamalia, M. (1989). Intentional learning as a goal of instruction. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Collins, A., & Ferguson, W. (1993). Epistemic forms and epistemic games: Structures and strategies to guide inquiry. *Educational Psychologist, 28*(1), 25-42.

Johnson, R.E. (1995) *A Model Computer: Algorithms for Machine and Assembly Programming.* Minneapolis: West Publishing Company.

Jonassen, D.H., Beissner, K., & Yacci, M. (1993). *Structural knowledge: Techniques for representing, conveying, and acquiring structural knowledge.* Hillsdale NJ: Erlbaum.

Lohse, G.L., Biolsi, K., Walker, N., & Rueter, H.H. (1994). A classification of visual representations. *Communications of the ACM, 37*(12), 36-49.

Mayerhofer, M.A., & de Lucena, C.J.P. (1992). Design of an algorithm simulation and animation environment. *SIGCSE Bulletin, 24*(2), 7-14.

Morrison, D., & Collins, A. (in press). Epistemic fluency and Constructivist learning environments. *Educational Technology Journal.*

Norman, D. (1993). *Things that make us smart: Defending human attributes in the age of the machine.* New York: Addison-Wesley.

Perry, W. (1985, May). Different worlds in the same classroom: Students' evolution in their vision of knowledge and their expectations of teachers. *On Teaching and Learning,* 1-17.

Petzold, C. (1990). Working with metafiles: Modifying and sharing vector-oriented images. *PC Magazine, 9*(17), 449-455.

Reigeluth, C.M. (Ed.) (1983). *Instructional-design theories and models.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Reigeluth, C.M. (Ed.) (1987). *Instructional theories in action.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Saettler, P. (1990). *The Evolution of American educational technology.* Englewood, CO: Libraries Unlimited.

Seels, B. (1995). *Instructional design fundamentals: A reconsideration.* Englewood Cliffs, NJ: Educational Technology Publications.

Sherry, L. (1995). A model computer simulation as an epistemic game. *ACM SIGCSE Bulletin, 27*(2), 59-64.

Trigg, M. (1995). *Creating and playing epistemic games: A teacher's guide to using knowledge generating activities in the classroom.* (Available from Maggie Trigg, CIS Department, Arapahoe Community College, 2500 West College Drive, Littleton, CO 80160-9002.)

Zernik, D., Snir, M., & Malki, D. (1992). Using visualization tools to understand concurrency. *IEEE Software, 9*(3),87-93.

[back arrow] Back to Scholarly Publications
[back arrow] Back to Lorraine's Publications

---

lsherry@carbon.cudenver.edu
Updated June 8, 1996