Supporting Educational Software Design with Knowledge-Rich Tools

Benjamin Bell
Department of Communications, Computing, and Technology in Education
Teachers College, Columbia University
bell@ilt.columbia.edu

1. Authoring Tools and GBS Builder

As educational software becomes more complex and multimedia more pervasive, authoring tools become increasingly central to the construction of instructional applications. Moreover, the importance of creating tools that encourage effective instructional design is now widely recognized. In response to this, researchers are creating sophisticated tools that not only support the mechanics of software design but embody models of pedagogy as well, which has led to efforts in designing specialized authoring environments around specific instructional frameworks. To better support the design process, such tools must possess knowledge of the domain and of the kinds of tasks suited to that domain. This paper reports preliminary work in augmenting a specialized authoring tool with task and domain knowledge. The tool supports the design of a specific class of Goal-Based Scenarios that emphasizes investigation. The extension to this tool represents and applies knowledge of investigations, in terms of both their structure and the objects that typically appear in them.

Goal-Based Scenarios (GBS) [Schank, Fano, Bell, & Jona 1994] is a framework for simulation-based learn-by-doing instruction, in which the learner is engaged in pursuing a goal, within a simulated environment, in order to master a set of target skills. The student is an active participant in the scenario, assuming a role in which resources provided by the program are available to help the student progress toward completing the task. There is a potentially wide range of programs that could be designed in accordance with GBS principles: programs that allow students to build artifacts; programs in which the student controls a device or participates in a process; programs in which the student conducts investigations, and so on. Because we are interested in creating tools that help designers create GBSs, it is important to design the right sort of tool for each class of GBS, of which Schank, Korcuska, et al. [1996] have identified eight.

A class of GBS called *Investigate & Decide* [Bell 1996] describes programs in which the principal focus is on conducting investigations to gather information in support of a decision the student is asked to make. The utility of the Investigate & Decide model would be amplified if the process of instantiating this class of GBS in software were made less onerous. This is the motivation behind an authoring environment called the Goal-Based Scenario Builder (GBSB) [Bell 1996]. GBSB is a prototype GBS construction tool, built collaboratively by a team of researchers and developers at the Institute for the Learning Sciences [1]. The purpose of the tool is to allow a domain expert or teacher to create an Investigate & Decide GBS, without requiring expertise in programming and instructional design. The tool employs the underlying model to govern the interaction, so that the design process is carried out in a manner consistent with the organization of the model. Adherence to the model helps a designer create a GBS that observes the principles embodied in the framework.

2. Investigation Knowledge and GBS Builder

The extent to which a tool can assist a designer is governed in part by the level of knowledge that the tool possesses about the artifact being constructed. In considering the leverage that a tool like GBSB provides, we must first ask what the tool "knows" about investigation and how that knowledge contributes to the tool's design support role. GBSB includes a specification of the investigation task as consisting of three phases: Obtain Sample, Analyze Sample, and Interpret Results. Within each phase, the model directs the interaction by defining which interface features are to be elicited and in what order. In Analyze Sample, for instance, design parameters elicited for running a particular test include the text to label the test button, a picture of the test

[1] Smadar Kedar and the author led the design effort, with guidance from Roger Schank, Chris Riesbeck, Ray Bareiss, and Alex Kass. Steven Feist and Erica Dubach contributed their programming talents; Jaret Knyal supplied the interface artwork. A proof-of-concept version of the tool, which preceded the prototype, was created by the author, Michael Korcuska, and Edward Lim.

device in its start state, that picture's caption, a picture of the test after it has been run, and that picture's caption. These elements, and their ordering, constitutes the model's "knowledge" of the structure of investigations. While this model is sufficient for guiding the designer in supplying all required interface parameters, it obviously limits the program's ability to assist in creating the investigation.

Empirical studies of GBSB [described in Bell 1996] suggest that users find the interface orientation of the tool to be of significant, but limited value. A common reaction among users was that the tool lacked a conceptual perspective, that is, the guidance it provides is strictly interface-oriented. While the model specifies in detail which objects must be defined, it does not specify much about the relationships among those objects or about the criteria for defining an object.

Why does this matter? The purpose of articulating a model in the first place is to embody a set of design standards that a tool could help a designer observe. By expressing such standards in terms of interface elements that a designer must specify, the tool has no basis from which to help a designer make intelligent decisions about how to define these elements. The question is what additional knowledge would the tool require in order to offer a useful measure of guidance in designing investigations. Two kinds of information are relevant here: structural knowledge and domain knowledge. Structural knowledge informs the designer that an investigation involves an analysis procedure, applied to some sample, and that the results of this procedure can contribute to some appropriate conclusion. Domain knowledge informs a designer that, for example, a refractometer can determine an unknown gem's index of refraction, or that gas chromatography can reveal the presence of substances used by arsonists to fuel a blaze.

3. Representational Primitives for Investigation Knowledge

Domain knowledge, that is, pertaining to the kinds of components likely to appear in investigations, includes knowledge of the components themselves (*e.g.*, that a thermometer measures temperature), as well as knowledge of the relationships among them (*e.g.*, that atomic absorption detects the presence of elemental metals). Both of these kinds of information can be sufficiently represented using a taxonomic frame representation, as in Figure 1.

Class: Microscope
Superclass: Cellular-test
Results: (cell-morphology)
Default pict empty: micro-empty.pict
Default pict full: micro-full.pict

Figure 1: Example definition for subclass of TEST

Structural knowledge is codified as an investigation *template*. A template contains four elements: sample, test, result, and outcome. A complete template specifies the types of objects which will be part of an investigation sequence. Each can be associated with specific *scenarios*. An investigation scenario supplies specific instances for the objects named in the template. If a template includes, for example, the outcome "cause of death", then a scenario for this template might contain the outcome "drug overdose". A scenario thus defines an explicit sequence of activity which a user of the resulting GBS would encounter.

4. Structuring and Manipulating Investigation Knowledge

The knowledge called for in the preceding section is encoded in an extension to GBS Builder, called the Investigation Map, or *IMap*. The IMap consists of an editor for adding elements to an investigation, an advisor, suggesting ways a designer might complete a partially-defined investigation, and a knowledge base for maintaining elements which designers may select for their investigations. A designer builds an IMap by creating an investigation template, and then defining scenarios which specify specific paths through the template. The input the designer provides to the IMap thus includes the samples, tests, results, and outcomes populating a particular investigation.

4.1. Defining an Investigation in IMap

To illustrate a typical interaction, let us consider an investigation in which the student must determine whether or not a lake has become polluted. The first step is to create a template which defines an appropriate type of investigation. The purpose of this investigation is to determine "cause of change in ecosystem", which the designer selects to occupy the "outcome" part of the investigation template. The next item to fill in is what kind of results will help determine a cause of change in ecosystem. The designer in this example intends for the student to use the level of dissolved oxygen in the lake as a source for this kind of evidence, and thus the "result" part of the template is filled with "concentration dissolved gas" [Fig. 2].

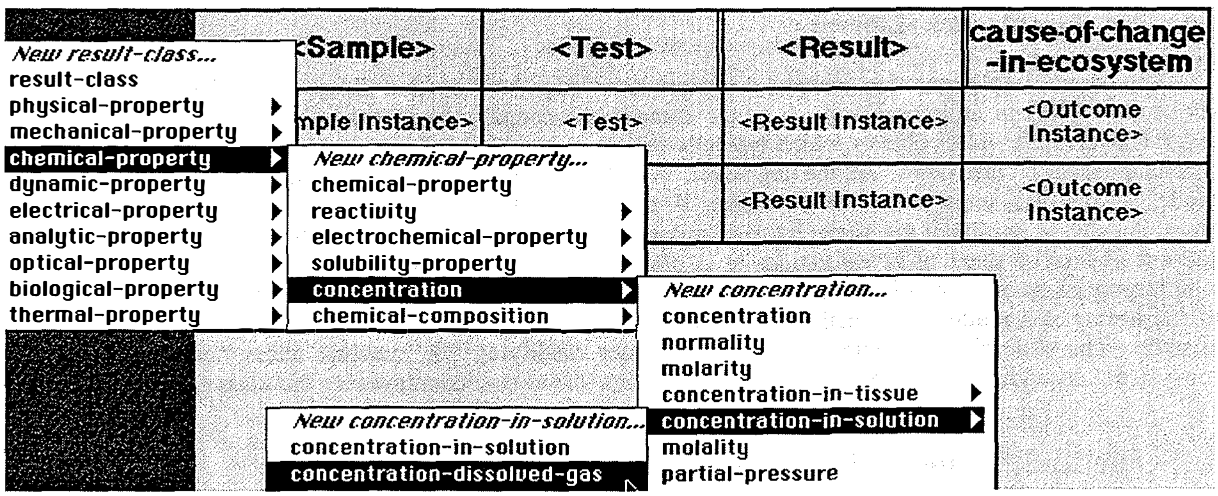

Figure 2: Selecting "concentration dissolved gas" as the investigation's result

The designer can then select the type of device which can measure this concentration (in this case, "dissolved oxygen test"), and the type of samples which will be tested ("water sample"). The completed template occupies the top row of the table in Figure 3.

Water Sample	Dissolved Oxygen Test	Concentration Dissolved Gas	Cause Of Change In Ecosystem
Southeast Region Sample	Dissolved Oxygen Test	3.8 Ppm	Critical Concentration
Southwest Region Sample	Dissolved Oxygen Test	4.3 Ppm	Low Concentration
Northwest Region Sample	Dissolved Oxygen Test	5.0 Ppm	Moderate Concentration
Northeast Region Sample	Dissolved Oxygen Test	6.5 Ppm	Normal Concentration

Figure 3: Example of a Completed IMap

4.2. Applying Investigation Knowledge for Design Support

As the designer creates the investigation (that is, the samples, tests, results, and outcomes in the template), IMap can offers suggestions about how to proceed. A suggestion contains two kinds of information. This first is syntactic advice, *i.e.* indicating which part of the template should next be defined. The second form of guidance which IMap offers is semantic, *i.e.* recommending specific kinds of objects which would be consistent with those already installed into the template. Figure 4 shows the guidance IMap offers to a designer who has specified an outcome of "cause of change in ecosystem".

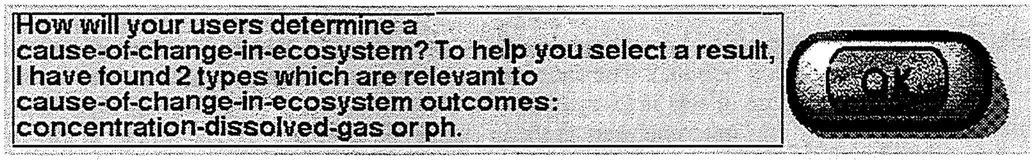

Figure 4: Template guidance for selecting a result class_

4.3. The Investigation Library

The current implementation includes a modest knowledge base so that IMap may offer design guidance about the objects in an investigation (*i.e.*, the domain knowledge). The investigation library currently defines several hundred classes of objects which typically appear in the kinds of investigations generally conducted in secondary science classrooms. At the top level, the classes are organized according to IMap's investigation model, as samples, tests, result, or outcomes. Within each category is a set of class definitions, organized hierarchically. A portion of the hierarchy from one such category is shown in Figure 5. Since the tool should support as broad a range of investigations as is practical, the interface includes a knowledge acquisition mode which supports the addition of new definitions and modification of the existing knowledge. Adding a new class to the library is a matter of indicating its superclass, and identifying the classes to which this new class is related. The example in Figure 6 shows a designer modifying the outcome class "cause-of-change-in-ecosystem", by adding "temperature" to the dependencies (list of results) relevant to this class of outcome.

5. Scope and Content of the Baseline Investigation Library

Determining what knowledge to include in IMap's initial knowledge based relied in part on the GBSs which had been constructed using GBSB (prior to IMap) [2]. The investigation elements included in those applications formed the beginnings of the knowledge base, which was expanded by generalizing these elements to form broader categories until the top-level categories (sample, test, etc.) were reached. Table 1 lists these GBSs and the kinds of investigation elements contributed by each. As the table suggests, this primitive model of investigation is likely to be sufficient for a reasonable range of potential applications.

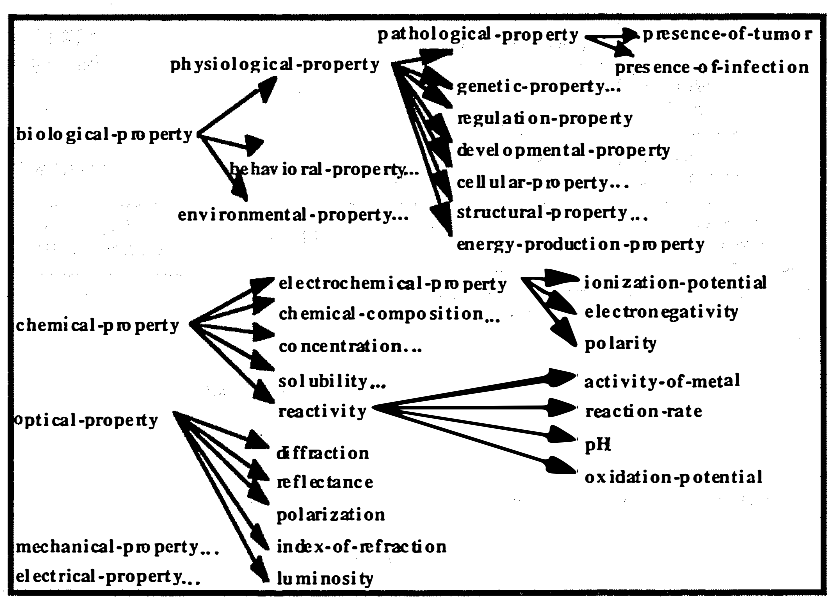

Figure 5: Partial hierarchy for result classes

[2] These GBSs were constructed by ILS graduate students and by teachers from a local school district [Bell 1996].

In concert with the analysis of the available GBSs, a survey of the experimental activities typically included in high school science curricula was conducted to guide the evolution of the knowledge base. Science laboratory supplements were used from Biology [Brown 1978; BSCS 1973a; BSCS 1973b; BSCS 1990; Edwards 1975; Kroeber 1965; Moore 1970], Chemistry [Ferguson 1978; Garrett 1966; McGill 1966; Metcalfe 1982; Sutman 1967], and Physics [Stollberg 1975; Taffel 1970; Williams 1973]. These texts supplied information which shaped the sample, test, result, and outcome hierarchies.

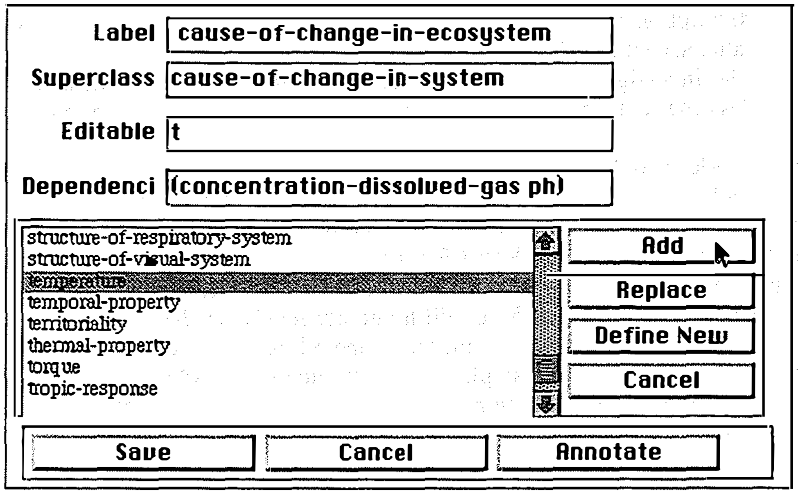

Figure 6: Defining a new outcome class

Title	*samples*	*tests*	*results*	*outcomes*
Arson Investigator	textile	gas chromatography	presence	cause of fire
Breast Cancer Counselor	breast tissue	mammogram, biopsy	malignancy, presence of tumor	diagnosis
Cardiac Counselor	human, blood	EKG, reflotron	frequency, composition	risk assessment
Crime Lab	liver, blood	immunoassay, atomic absorption	concentration of toxin in tissue	cause of death
EPA Advisor	water	assay	presence	cause of change in ecosystem
Gem Detective	crystal	refractometer, microscope	structure, index of refraction	artifact type
Pollution Investigator	water	pH Meter, dissolved oxygen test	pH, concentration dissolved gas	cause of change in ecosystem
Public Health Administrator	water	pH, spectrometer	concentration in solution	cause of illness
Sickle Cell Counselor	blood	electrophoresis, microscope	genetic makeup	relative probabilities
Storm Watcher	air	radar, barometer	air pressure, range & bearing	predicted location

Table 1: Completed GBSs within the scope of IMap's knowledge base

6. Conclusion

IMap is an extension to GBS Builder for supporting creation of investigations in Investigate & Decide GBSs. The IMap tool is a knowledge-rich editor, allowing a designer to instantiate investigation elements, and also a help utility, able to make suggestions to ensure that the designer's investigation is consistent with its internal model. Motivating IMap are the obstacles typically faced in creating an investigation. This knowledge includes both a general component (a model of investigation), and specific knowledge about the elements within an investigation.

IMap is a useful design aid because designers are more adept at imagining appropriate investigations than at conceiving of all the necessary details [3]. For example, a chemistry teacher may express a GBS design as "students will learn about oxidation by being arson investigators, looking for clues by testing combusted materials." What is likely to pose more difficulty is turning that idea into a working system. IMap can assist designers who start with a broadly-framed idea by helping operationalize that idea as an investigation, turning the designer's general description into an explicit set of objects and sequence of interactions among those objects. Another, pragmatic, aspect to IMap's utility is that it can reduce the effort needed to create the interface screens associated with the investigation. The knowledge representation in IMap includes default graphics and labels associated each class that are automatically inserted into the designer's investigation at the appropriate locations in the interface.

One area of future work is to broaden the representation of investigation elements, so that IMap could supply answers to standard questions about parts of an investigation, for example "what is this?", or "what accounts for property X?". Another area for future work is in reusing investigations. While IMap currently allows any class listed in the library to be selected for inclusion in an investigation, a more powerful facility would allow for complete investigations to be retrieved that match the designer's objectives according to some appropriate similarity metric. The utility of IMap will hinge ultimately on the scope of its knowledge base, so knowledge acquisition tools that go beyond the current frame editors will be required. Finally, more complex models of investigation will be needed. The sample-test-result-outcome paradigm is a convenient simplification, but IMap will need to accommodate investigations in which, for example, two tests provide conflicting results, or an outcome depends on multiple tests. The implementation reported in this paper establishes an architecture within which these extensions may be readily introduced.

References

Bell, B.L. (1996). A special-purpose architecture for the design of educational software. Ph.D. Dissertation, The Institute for the Learning Sciences, Northwestern University, Evanston, IL.

Brown, W.H. (1978b). *Concepts and Inquiries in Biology.* Educational Methods.

BSCS (1973a). *Biological Science: an inquiry into life.* Harcourt, Brace, Jovanovich: New York.

BSCS (1973b). *Biological Science, molecules to man.* Houghton Mifflin: Boston, MA.

BSCS (1990). *Biological Science, a molecular approach.* Heath & Company: Lexington, MA.

Edwards, G.I. and Cimmino, M. (1975). *Laboratory Techniques for High Schools.* Barron's Educational Series: Woodbury, NY.

Ferguson, H.W., Schmuckler, J.S., Caro, A.N., and Johnson, A. (1978). *Laboratory Investigations in Chemistry.* Silver Burdett: Morristown, N.J.

Garrett, A.B., Richardson, J.S., and Montague, E.J. (1966). *Laboratory and demonstration problems for Chemistry.* Ginn & Company.

Kroeber, E., Wolff, W.H., and Weaver, R.L. (1965). *Biology.* Heath & Company.

McGill, M.V., Bradbury, G.M., and Sigler, E.A. (1966). *Chemistry Guide and Laboratory Activities.* Lyons & Carnahan: Chicago.

Metcalfe, H.C., Williams, J.E., and Castka, J.F. (1982). *Exercises and Experiments in Modern Chemistry.* Holt, Rinehart, & Winston: New York.

Moore, H.A. and Carlock, J.R. (1970). *In the Laboratory: the Spectrum of Life.* Harper & Row: New York.

Schank, R.C., Korcuska, M., and contributors (1996). Eight Goal-Based Scenario Tools. Technical Report 67, The Institute for the Learning Sciences, Northwestern University, Evanston, IL.

Schank, R.C., Fano, A., Bell, B.L., and Jona, M.Y. (1994). The Design of Goal Based Scenarios. *The Journal of the Learning Sciences*, *3*(4), 305–345.

Stollberg, R. and Hill, F.F. (1975). *Physics fundamentals and frontiers.* Houghton Mifflin: Boston, MA.

Sutman, F.X., Harris, S.P., and Greenstone, A.W. (1967). *Concepts in Chemistry: Teacher's manual and answer key.* Harcourt, Brace & World.

Taffel, A., Baumel, A., and Landecker, L. (1970). *Physics Laboratory Manual.* Allyn & Bacon: Boston, MA.

Williams, J.E., Trinklein, F.E., Metcalfe, H.C., and Lefler, R.W. (1973). *Laboratory Experiments in Physics.* Holt, Rinehart, & Winston: New York.

[3]an observation due to Chris Riesbeck