Supporting Educational Software Design with Knowledge-Rich Tools

Benjamin Bell
Center for Intelligent Tools in Education
Teachers College, Columbia University

benjamin.bell@columbia.edu

Address correspondence to:

Benjamin Bell
Box 8
Teachers College, Columbia University
525 W. 120th Street
New York, NY 10027

(212) 678-4113
benjamin.bell@columbia.edu
Abstract

Several questions have emerged surrounding the design of authoring tools for instructional software that have helped frame an on-going dialog within the community. One such question is how specific or customized an authoring tool should be with respect to the range of applications it can support. One project that comes down on the side of specificity is IDLE-Tool, which guides authors through the process of creating an Investigate and Decide Learning Environment. Despite the narrow focus, though, the original instantiation of this tool lacked any real knowledge of the investigation process and its components. An add-on to this tool supplies an Investigation Map (Imap) that brings a richer representational scheme to the underlying tool. This article summarizes the IDLE-Tool work and introduces a set of primitives that capture the structure of a simplified form of investigation. The functioning of the tool after being enhanced with Imap is presented in detail, and a limited knowledge base of investigation components is described.
Life with a knowledge-rich authoring tool

The process of creating sophisticated instructional applications is costly, complex and tedious and relies on expertise from multiple disciplines. Authoring tools are applications that aim to reduce the effort needed to produce software, by assuming responsibility for mechanical aspects of the task, by guiding the author, and by offering predefined elements that an author can package together to suit a particular need.

Ideally, such a tool would have some understanding of what the author wishes to create and could then offer more useful and specific support. Consider a category of learning environments in which the student conducts an investigation and makes some decision on the basis of his findings. To author such a learning environment requires some understanding of the structure the investigation (as a learning environment) should follow, so that students could potentially develop appropriate decisions based on information derived by applying some testing, measuring, or analytic procedures. The author would also need to possess some understanding of the domain, so that appropriate materials, apparatus, and so on would populate the investigating and deciding activities.

Consider an Earth Sciences teacher designing a simulation in which the student must determine the source of a contaminant that is polluting a lake. Where to begin? To generate a cohesive environment requires that the author systematically consider all the relevant variables that could enter into the
student’s inquiry and what predictions the student might make based on the values that these variables assume.

Alternatively, the authoring tool could itself could offer a template that lays out the appropriate parameters that the author needs to define. For instance, it is important to consider what the student will be deciding (what caused the change in the ecosystem?) and how the student will make this determination (measuring dissolved gases? pH?). Suppose the Earth Sciences teacher in this example wanted to browse around for suggestions as to how students could measure the level of oxygen in the lake water. We would then want a tool that allows the author to view hierarchically the relevant portions of the knowledge (Figure 1).

![Figure 1. Selecting a result from the knowledge base during investigation design](image)

To enhance the power of this tool it should also be equipped to recommend some appropriate testing procedures (Figure 2).
Supporting Educational Software Design with Knowledge-Rich Tools

Figure 2. Suggestions from the knowledge base offered by tool

To help the author maintain orientation, the tool should also present a summary view of the investigation that the author has specified thus far (top row of Figure 3).

<table>
<thead>
<tr>
<th>Water Sample</th>
<th>Dissolved Oxygen Test</th>
<th>Concentration Dissolved Gas</th>
<th>Cause Of Change In Ecosystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Southeast Region</td>
<td>Dissolved Oxygen Test</td>
<td>3.8 Ppm</td>
<td>Critical Concentration</td>
</tr>
<tr>
<td>Sample</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Southwest Region</td>
<td>Dissolved Oxygen Test</td>
<td>4.3 Ppm</td>
<td>Low Concentration</td>
</tr>
<tr>
<td>Sample</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Northwest Region</td>
<td>Dissolved Oxygen Test</td>
<td>5.0 Ppm</td>
<td>Moderate Concentration</td>
</tr>
<tr>
<td>Sample</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Northeast Region</td>
<td>Dissolved Oxygen Test</td>
<td>6.5 Ppm</td>
<td>Normal Concentration</td>
</tr>
<tr>
<td>Sample</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. Example design template for lake pollution example

In addition the author must instantiate specific paths or scenarios. To accommodate the case in which a specific region of the lake possesses a low level of dissolved oxygen, for example, the author might define the scenario shown in the second row of the table in Figure 3.

This brief example is a preview of what a tool with limited but specific knowledge of the author’s task can offer in the way of design support. But what
knowledge must an authoring tool possess in order to offer such guidance? The issues of what (and how much) knowledge is required, how it should be represented, and how it can brought to bear on the authoring process are the questions addressed in this article. The first part of this article summarizes an authoring tool that lacks knowledge of the task semantics and the domain, but instead relies on a syntactic model. The remainder of the article introduces an extension to this tool that supports the author with specific structural and domain-level knowledge, and provides detailed examples of how this knowledge contributes operationally during the authoring process.

Investigate & Decide Learning Environments

This article presents a knowledge-rich extension to an authoring tool created to aid in the construction of Goal-Based Scenarios (Schank, Fano, Bell, & Jona 1994). Goal-Based Scenarios (GBS) 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.

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 (1996) have identified eight.
A class of GBS called *Investigate & Decide Learning Environments* (Bell, 1998) describes applications in which the principal focus is on conducting investigations to gather information in support of a decision the student is asked to make. Investigate & Decide Learning Environments (IDLE) is a model for instruction aimed at teaching the conceptual and causal elements of a specific domain or set of related phenomena, in which the learner is engaged in an inquiry process based in analysis (investigation) and synthesis (decision-making).

Specializing the GBS model for Investigate & Decide learning is useful in two ways. First, it allows an instructional model to be defined that adheres closely to teaching principles and strategies. Second, an authoring tool that embodies this model can become a more powerful design assistant, since it can “know” more about the objectives and constraints that characterize the author’s design activities (Bell, 1995). In other words, the design constraints the tool is meant to enforce become easier to operationalize when they are expressed in more specific terms. Categories like IDLE thus represent abstractions useful for developing a suite of specialized GBS tools.

**IDLE-Tool: A not-so-knowledge-rich authoring tool**

*Summary*

IDLE-Tool 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
IDLE application, 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.

IDLE-Tool steps the author through the design process in a protocol described as Guided Case Adaptation (Bell, 1998), which blends case-based design, hierarchical refinement, and generate-and-test modes of interaction. A guided case adaptation session allows an author to incrementally alter an exemplar application until it becomes a new IDLE application that satisfies the author’s original design criteria.

The interface includes a workspace, in which the designer’s emerging GBS is shown and in which the example program, Sickle Cell Counselor (Bell, Bariess, & Beckwith, 1994) is also displayed on request. A graphic representation of the GBS model highlights the part of the model that the designer is currently working on, and serves also as control buttons that can be used to jump to a desired part of the model. The tool operates in three modes. Build mode carries out the guided case-based adaptation. Run mode allows the designer to try out the current GBS as defined thus far in the interaction. Design mode enables Investigation Mapping, which is the focus of this article.

**Field Testing**

Preliminary studies were conducted with three groups of authors who used IDLE-Tool to create Investigate & Decide Learning Environments (Bell,
During the early implementation phase of this work, twenty-one first-year graduate students participating in a seminar were assigned the task of creating GBSs using the tool. The objective was to collect early user data to help guide the design of the tool. Ten GBSs were created using IDLE-Tool. In general, students reported that they liked the basic structure of the tool and that using it yielded a speedup during the initial design phase. A common suggestion (50%) was that the tool needed to provide guidance that was more conceptually-oriented and less interface-oriented. The tool used in the experiment guided a step-by-step adaptation of the exemplar GBS, but did not attempt to explain the overarching concepts governing these steps. This interface guidance fell short of the kind of conceptual guidance the subjects said they wanted to see, with the reported consequences including losing focus on content design, and losing track of the overall flow because editing proceeds one screen at a time.

A second formative evaluation was performed with classroom teachers, several months after the conclusion of the first. Eight primary school teachers from a nearby district were chosen by a school official from among a self-selected applicant pool. The teachers were given an overview of Goal-Based Scenarios and a demo of IDLE-Tool, worked in pairs half-time over a six-week period. Four Goal-Based Scenarios were created using the tool. The most obvious difference between these GBSs and those generated by the graduate students was that, in general, the teachers’ GBSs were qualitatively better from an instructional perspective. That is, each of the teachers’ GBSs possessed a well-defined set of pedagogical goals which the scenario addressed in a
coherent way. All design teams reported that they liked IDLE-Tool and found it
useful as a design tool. All teams also reported difficulties with structuring the
investigation. One team drew a flowchart and said that it helped them determine
what information would be important to students and how students could derive
that information. Another team described having difficulty "zeroing in on (the
investigation)".

Nine months from the conclusion of the second study, a third evaluation
was conducted using first-year graduate students enrolled in a seminar. As part
of the seminar, students were shown four GBS design tools (including IDLE-
Tool), and asked to form into pairs and select one of the tools to use in creating
a GBS. Eight of the eighteen students elected to use IDLE-Tool. After three
weeks of design work, each team demonstrated their completed GBSs. The
GBSs produced in this study all followed closely the Investigate & Decide model.
This was the most obvious contrast between the GBS from this group of subjects
and those of the graduate students in the first study. The most likely reason for
this is that these groups selected IDLE-Tool from among four tools. They
therefore chose this tool because it represented the best match between their
preliminary designs and the various tools [2].

Discussion: Lessons Learned

Table 1 summarizes the IDLE applications created in the three trials.
Results from these three trials offered preliminary support for the IDLE-Tool as a
useful authoring environment, based on the relative ease with which authors in
all three trials constructed IDLE applications, and on the short periods of time in which they executed their designs.

**Table 1. GBSs created by subjects in the three trials**

<table>
<thead>
<tr>
<th>Title</th>
<th>Mission</th>
<th>Trial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aztec, P.I.</td>
<td>advise curator about objects’ authenticity</td>
<td>1</td>
</tr>
<tr>
<td>Cardiac Counselor</td>
<td>advise clients at risk for heart disease</td>
<td>1</td>
</tr>
<tr>
<td>Crash Course</td>
<td>determine driver at fault in traffic accident</td>
<td>1</td>
</tr>
<tr>
<td>Diver Down</td>
<td>help plan scuba dives</td>
<td>1</td>
</tr>
<tr>
<td>Environmental Decision Maker</td>
<td>take action at a polluted site</td>
<td>1</td>
</tr>
<tr>
<td>GBSex</td>
<td>counsel clients about contraception</td>
<td>1</td>
</tr>
<tr>
<td>Public Health Advisor</td>
<td>advise mayor about outbreak of illness</td>
<td>1</td>
</tr>
<tr>
<td>Rags to Riches</td>
<td>advise investors</td>
<td>1</td>
</tr>
<tr>
<td>Scandal!</td>
<td>draft news story about a scandal</td>
<td>1</td>
</tr>
<tr>
<td>Secret Caves of Ellora</td>
<td>find solution to archeological problem</td>
<td>1</td>
</tr>
<tr>
<td>Debt Buster</td>
<td>advise teens about personal financial management</td>
<td>2</td>
</tr>
<tr>
<td>Kid Counselor</td>
<td>help other kids resolve interpersonal conflicts</td>
<td>2</td>
</tr>
<tr>
<td>Lend a Hand</td>
<td>help disabled kids negotiate everyday activities</td>
<td>2</td>
</tr>
<tr>
<td>Arson Investigator</td>
<td>advise fire chief about causes of suspicious fires</td>
<td>2</td>
</tr>
<tr>
<td>Breast Cancer Counselor</td>
<td>test patients for Breast Cancer and explain treatments</td>
<td>3</td>
</tr>
<tr>
<td>Gem Detective</td>
<td>advise jeweler about identity of gem samples</td>
<td>3</td>
</tr>
<tr>
<td>Pollution Investigator</td>
<td>recommend action to reduce pollution in lake</td>
<td>3</td>
</tr>
<tr>
<td>Storm Watcher</td>
<td>advise theme park as to whether storm will hit park</td>
<td>3</td>
</tr>
</tbody>
</table>
Of course, subjects were not entirely satisfied with the tool. The most common reaction and the one with the most central implications is that the tool focuses too much on the details of the interface and too little on the underlying design principles. The guidance it provides is interface-oriented: the tool steps the designer through the process of specifying all elements of the interface called for by the model. While this 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. A way to address this is to modify the tool to adopt a conceptual orientation to supplement its interface orientation. For IDLE-Tool, this means that the tool should be able to represent the elements of investigating and deciding so that the designer would be specifying, not only interface parameters, but conceptual elements such as the source to be analyzed, the device performing the analysis, what information is derived, and what the student should be deciding based on that information. In order to support the design of GBSs at this conceptual level, the tool must be augmented with specific knowledge about the tasks implicit in the model (i.e., how investigating and deciding tasks are structured and what elements are needed).

Related Approaches to Authoring Tools

From General-purpose to Courseware Authoring

A number of sophisticated tools have emerged for creating interactive, multimedia software, including commercially available products such as
IconAuthor and Authorware. These "general-purpose" tools serve a variety of functions, but offer little in the way of design constraints governing the kind of software which can be produced. The result is a tool which supports a broad range of possible applications, some of which may be good, and some of which are likely to be poorly executed, but none of which will have been created with much guidance from the tool itself.

For a more specialized purpose, for example, the creation of instructional software, tools can be constructed which are based on more specific models (Macmillan, Emme, & Berkowitz, 1988; Murray & Woolf, 1992; Russell, Moran, & Jordan, 1988). The basic idea of these "ITS tools" is to enable non-programmers (e.g., a classroom teacher) to create instructional computer programs by setting values within some predefined template. Tools for the design of educational software, though going a long way toward a truly useful tool, still aim to support the creation of any possible kind of instruction. In doing so, they base the interaction around general models of instruction, which unfortunately are too general to serve as a specification for a piece of educational software, and are too general to be of much help in guiding a designer in creating such software, for three reasons.

First, there may be a wide range of categories of instructional software; ITS tools have typically been based on prototypes in no more than a few of these categories, so understanding what a universal tool should do, exactly, is not yet possible. Second, the enormous range of potential activities which a general tool would need to support would require a detailed knowledge of the structure of
activities and of the elements participating in these activities. Third, the guidance the tool provides would, by necessity, be in general terms so that the tool could offer help in any design context. The consequence here is that the value of the tool as an intelligent critic would be sacrificed in the name of generality.

**From Courseware Authoring to Knowledge-Rich Tool**

More recently, researchers have begun to address the need for authoring tools that possess specialized knowledge about their domain. Some approaches rely on an expert systems substrate (Reinhardt, 1997) or on a case-based reasoning engine (Goan, Stottler, & Henke, 1997) for capturing and representing such knowledge. The knowledge itself is seen by some as a modular component in an authoring tool that can be assembled from appropriate parts (Ritter & Blessing, 1998), though for some researchers, a modular architecture raises the concern mentioned previously that generality results in weak instructional models (Jona & Kass, 1997). A superior approach is to develop a suite of stand-alone specialized tools that collectively cover the same range of authoring needs (Ibid). Besides domain knowledge, researchers are also calling for pedagogical knowledge to be more explicitly incorporated into authoring environments (Jona, 1995; Major, Ainsworth, & Wood, 1997; Murray, 1998). The results reported in this article represent an attempt to accommodate the need for explicit pedagogies by supporting authors in creating a specific mode of instruction (Investigate & Decide Learning Environments). This work also recognizes the need for representation of domain knowledge. The ways in which these needs are addressed are discussed in the next section.
Blueprint for a smarter tool

To transform IDLE-Tool into a more intelligent design aid, we must decide what aspects of investigation design we would like an enhanced tool to support. We need also to determine what knowledge would be required to supplement the tool. Finally, we must invent a representation that captures that knowledge.

The internal structure of an investigation

Investigations can take many forms, but as used here, the term “investigation” takes on a specific meaning, namely, analyzing samples to derive data useful to support a particular hypothesis (Figure 4).

Figure 4. Investigation model for IDLE applications

The steps articulated in the model serve twin purposes. First, they provide a framework within which a designer can operationalize the design, by
decomposing it into subtasks of the investigation. Second, they provide a default behavior for the way the program will execute. For example, any device specified by the designer in the Analyze Sample phase will accept a sample, and apply a default procedure to that sample when activated (*i.e.*, it instantiates a default device). These default behaviors have both a graphical component, specified as a set of images provided by the designer (*e.g.*, “the machine in its empty state; the machine with a blood sample”), and an executable component defined by the program (*e.g.*, when the on-switch is pressed, show the machine-with-sample picture and mark the sample “tested” by this device). Within the investigation activity, the student not only synthesizes knowledge, but also makes decisions regarding what sort of knowledge to construct and how to interpret it. Moreover, the results a student gathers must guide him toward making the decision around which the scenario is based. This means that a designer’s specifications have an important semantic component extending beyond the investigation itself.

**Example interaction for investigation design**

The context for the following example is a dialog between the (original) prototype tool and a designer in the process of creating an IDLE application called *Crime Lab*, in which students act as forensic investigators to establish whether lethal levels of toxic material are present in tissue samples collected in an autopsy [3]. The designer’s objective is to implement a scenario in which an overdose of barbiturates is implicated. The tool asks the designer to specify various features of this scenario, eliciting text labels and graphics to be included
in the screens comprising the investigation. The excerpts below show, for a few representative elements of the Investigation, the way the tool prompts the designer:

Edit the button label for the source of the first sample

Take Liver Sample

Choose a picture of an empty container for the first sample

test-tube-empty.pict

Choose a picture for a full container for the first sample, which will appear after the student clicks on the extract sample movie.

test-tube-full.pict

Edit the button label for the first test. When the student clicks on this button, a graphic of the first test will appear.

Run Immunoassay Test

Figure 5 shows a screen resulting from the example dialog. The new Investigation reflects the parameters specified by the designer, supported by the default behavior of the executable GBS. Note, however, that the tool operates under the assumption that the designer possesses sufficient knowledge so that the resulting investigation is both internally consistent and relevant to the overall mission of the GBS. The tool enforces only syntactic constraints on which objects and labels are specifiable, and functional constraints on their run-time behavior.
What the tool lacks is a representation of the designer’s input choices. For example, the tool requires the designer to supply a label for the button which selects sample number one; the tool is also capable of processing clicks from that button. But the tool does not reason about the sample which that button selects; the designer’s input here is treated simply as a text string. The decisions about which elements to include in an investigation are based on knowledge coming from the designer, not from the tool. The next section proposes a way to begin to classifying such knowledge.
Knowledge applied in creating an investigation

Looking more closely at how a designer constructs an investigation, we can identify two categories of knowledge being applied: domain knowledge of the elements in the investigation and their relationships, and structural knowledge of how a coherent investigation should flow. Domain knowledge is exercised when a designer selects specific elements for inclusion in an investigation. In the example dialog, selecting Immunoassay as a test could be appropriate because it is useful in detecting the presence of various drugs, which is central to the scenario that the designer is implementing. Liver tissue is useful because concentrations in the human liver at which various drugs are lethal are readily available. This kind of detailed knowledge represents a potential design obstacle. It is one thing to know the term “immunoassay”, but something else altogether to understand that process well enough to fill in the details of an investigation employing this process.

Structural knowledge helps guide the application of domain knowledge. In selecting immunoassay (which measures concentrations of a target substance), the designer in the example recognized that the possible outcomes the student will be attempting to establish should be distinguishable on the basis of results derived from the available tests. The designer further recognized, perhaps implicitly, that tests (1) are useful insofar as they derive necessary information, and (2) must be consistent with the samples available to be tested. The connection between the information the student is ultimately deriving, and what
his goal in the GBS is, can too easily be lost in the details of designing the investigation activities. Controlling the design by applying structural knowledge is another aspect of the design process.

The strategy here is to understand what it is designers know that enables them to create investigations. We have seen two kinds of knowledge that help a designer create an investigation: knowledge about the elements of a specific investigation, and knowledge about the structure of investigations in general. How is this distinction relevant to augmenting the capabilities of IDLE-Tool? Ideally, a tool would be able to critique a proposed investigation as well as offer suggestions for improving it. Identifying the structural and domain-specific aspects of investigation provides a starting point for considering what sort of knowledge is needed to move the prototype tool in this direction.

Investigation knowledge in the model

Now that we have a framework for analyzing investigation knowledge, we can revisit the knowledge implicit in the prototype tool. The following sections articulate what knowledge is there and consider where this knowledge is deficient in light of the previous discussion.

What the tool knows about investigations
The IDLE model treats investigation as consisting of the Obtain Sample, Analyze Sample, and Interpret Results phases. Within each phase, the model directs the interaction with the designer by defining which features are to be elicited and in what order. In Analyze Sample, for instance, the sequence of design parameters elicited for a running a particular test is as follows:

- test button label
- test empty pict
- test empty caption
- start test pict
- test full pict
- test full caption

The elements in the above list, 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 parameters, it obviously limits the program's ability to assist in creating the investigation, as discussed next.

**Limitations of the model**

To understand the limits of the current model, we can look at two examples of design constraints that a designer would typically observe in creating an investigation. The first design constraint is that the available tests should be able to yield the desired results. If the student is supposed to determine whether a victim died from a drug overdose, for example, atomic absorption would be an inappropriate test, since it is effective for detecting
elemental metals, but is not suitable for detecting barbiturates. The second constraint is that results obtained must relate to an outcome the student is investigating. If a student’s mission in a GBS were to establish whether mercury poisoning were the cause of death, for instance, results from a pH test would be irrelevant, since the acidity of a sample does not indicate the level of metals it contains.

Representing investigation knowledge

We now see that IDLE-Tool needs two kinds of knowledge: domain knowledge of the parts of an investigation, and general knowledge of the structure of investigations. In this section I discuss how these types of knowledge can be represented in IDLE-Tool.

Domain knowledge of investigation elements

Knowledge of 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.
The top-most level of this taxonomy identifies four basic types of investigation elements: the samples to be examined, tests to analyze these samples, results which could potentially be found, and the possible outcomes a user could identify. Figure 6 shows an example class definition for a subclass of Test.

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

**Figure 6.** Example definition for subclass of TEST

The definition shown in Figure 6 indicates that an instance of the class “microscope” can provide information about the morphology of a cell. Note the “Results” slot, which identifies which features a test is capable of measuring. Results are also associated with particular sample classes. The results “cell-morphology” and “Hg-genotype”, for example, are associated with the class “Blood-sample”, as shown in Figure 7.

```
Class: Blood-sample
Superclass: Circulatory-system-sample
Results: (cell-morphology Hg-genotype)
Default pict: blood-cells.pict
```

**Figure 7.** Example definition for subclass of SAMPLE
The results serve to identify the information derived from performing a given test on a given sample. For example, since the result “cell-morphology” appears in the definitions of “blood-sample” and “microscope”, any microscopic analysis of a blood sample could yield information about the shape of the cells in that sample. The definition of this result class is given in Figure 8. The information supplied in these definitions is applied during the design process as explained later in this article.

<table>
<thead>
<tr>
<th>Class: Cell-morphology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Superclass: Cellular-property</td>
</tr>
</tbody>
</table>

Figure 8. Example definition for subclass of RESULT

The examples presented so far represent domain knowledge of specific investigation components. In the next section, I show how the second type of knowledge, that of the general structure of investigations, may be represented.

Capturing structure via Investigation Templates

Investigations have a common sequence of scenes, partially captured by the phases of the Investigation in the current model. But the earlier discussion of how a designer builds an investigation implicated background knowledge beyond the Obtain-Analyze-Interpret phase definitions. For example, a designer looks for
(1) samples possessing features which could be useful in arriving at an outcome, and (2) tests appropriate for uncovering such features from certain classes of samples. That is, certain default expectations regarding the elements of these scenes helped guide the designer in filling in the investigation.

The general knowledge which a designer applies to create an investigation, therefore, can be succinctly codified as a set of scenes and expectations, packaged by an investigation template. A template contains four elements: sample, test, result, and outcome. A complete template specifies that the outcome the student is attempting to support could in theory be established using the materials available, that is, it makes the assertion that “Sample W, when analyzed using Test X, yields Result Y, which supports Outcome Z.”

Each template 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.

To illustrate, consider the example template and scenarios in Figure 9. Verifying the constraints associated with the investigation sequence is straightforward because the knowledge needed to do so is basically syntactic. The second scenario in the figure, for example, is incomplete because no result has been specified. Verifying that a particular template satisfies the expectations associated with an investigation, though, requires semantic information about the
items referred to in the template. The representation of such knowledge was introduced previously. Later, I describe a mechanism for applying this knowledge in helping a designer create a coherent investigation.

<table>
<thead>
<tr>
<th>blood sample</th>
<th>immunoassay</th>
<th>concentration</th>
<th>cause of death</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angela blood</td>
<td>immunoassay</td>
<td>lethal dose arsenic</td>
<td>poison</td>
</tr>
<tr>
<td>Victor blood</td>
<td>immunoassay</td>
<td>&lt;concentration&gt;</td>
<td>overdose</td>
</tr>
</tbody>
</table>

**Figure 9.** Example investigation template and associated scenarios

**Representational Issues for Investigation knowledge**

Three representational issues must be raised before proceeding. One issue is what leverage the investigation templates offer. The notion of packaging scenes and their default expectations within a structure is explained in detail by MOP theory (Schank, 1982). The MOP architecture provides a convenient structure for representing the internal configuration of investigations, and is well-suited to reasoning (part of Schank’s intent), which is useful from the standpoint of creating a tool which can act as an intelligent assistant. Moreover, it supports the storage and retrieval of structures in memory, which offers the potential to support the design process with a well-indexed library of investigations.

A second issue is how broad the representation is, that is, what range of investigations the representation can cover. The purpose of extending the tool is to support the design of activity in Investigate & Decide GBSs, so the functional criterion here is whether investigations in the I&D context can be expressed in
the vocabulary of the templates. One indication that the representation is sufficient would be its ability to capture the activities in the ten I&D GBSs that have been constructed to date.

Third, adopting a particular representation also raises the question of how a user would interact with the knowledge. In particular, we need some assurances that the representation is intuitive to a designer. A useful analogy to designing an investigation is creating an explanation (the outcome of an investigation is, in some sense, an explanation). In the course of generating an explanation, reasoners pose various explanation questions (Schank, 1986), for instance “what theory of physical causes explains this event?”, or “what circumstances led to this event?”. An investigation poses these same questions to the student, so the representation (in particular, the template’s “outcome”) mirrors the way a designer might think about the investigation. The remainder of the template is also structured around investigation questions that would naturally arise during design, namely, “what evidence would suggest this outcome?”, “what analysis procedure could supply this evidence?”, and “what source is subjected to this analysis?”, corresponding respectively to the result, test, and sample.

**Operationalizing investigation knowledge**

The preceding analysis of investigation knowledge and how it may be represented sets the stage for an extension to IDLE-Tool, 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.

**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. In thinking through the investigation, it is important to consider first what the student will be deciding, that is, what purpose the investigation serves. In this example it is to determine “cause of change in ecosystem”, which then becomes the “outcome” part of the investigation template. The next question to answer is how the student will make this determination, or put another way, 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”. 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.

Once a template has been specified, the designer can instantiate different scenarios to be included in the GBS. To accommodate the case in which a region of the lake possesses a low level of dissolved oxygen, for example, the designer creates the scenario shown in the second row of the table in Figure 3. This process can be repeated for creating additional scenarios for this template. The designer can also define new templates, for example, one in which the student measures the pH of the water to look for high acidity.

There are two points to note in this example. First, the designer is operating at a conceptual level, selecting classes of objects to include in the investigation. In contrast, recall the emphasis on interface issues characteristic of the earlier IDLE-Tool prototype. Second, the knowledge which the tool encodes is applied in supplying object classes to the designer, in this case, “cause of change in ecosystem”, “concentration dissolved gas”, etc.

Template Guidance

As the designer creates the investigation Imap can offer suggestions about how to proceed. A suggestion contains two kinds of information. This first is syntactic advice, indicating which part of the template should next be defined (according to the outcome-driven order in which Imap elicits the investigation). The second is semantic, recommending specific kinds of objects which would be
consistent with those already installed into the template. The suggestions which Imap offers at any point in the template construction are focused on the template item which Imap is currently expecting. For example, a designer who supplies an outcome and asks for help will be offered suggestions about which results might be appropriate (Figure 2). The suggestions Imap generates represent the choices which it knows to be compatible.

**Class Compatibilities**

Compatibility between two classes means that one class identifies another as being related in a particular way. A sample class and result class can be related via a *yields* link (top line in Figure 10). A test class and result class can also share a yields relation (middle line in Figure 10). A result class and outcome class are linked via a *supports* relation (bottom line in Figure 10). A sample class and test class are implicitly consistent if they share a common result class (small arrow in Figure 10).

![Figure 10. Link types among investigation classes](image)
Links are dynamic structures which serve to support IMap’s search for compatible classes, but are defined implicitly in the class descriptions within IMap’s knowledge base. For links between a sample class and result class, for example, IMap would retrieve the sample class’ definition and read the results slot. If sample class A possesses result B, the program asserts a yields link between sample class A and result class B. Appendix A defines the link types used in Imap's knowledge representation.

The investigation library

The architecture behind IMap calls for specific domain knowledge to be used in assisting a designer, so the baseline version of IMap must include a process for manipulating this knowledge, and some data to be manipulated. The initial library is aimed at investigations appropriate for scientific experiments in secondary physics, biology, and chemistry. The information within the libraries is aimed at helping the designer construct coherent investigations and at easing construction of the Investigation interface.

Scope of the baseline Knowledge Base

The investigation library 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 (example class
definitions were illustrated in Figures 6 — 8). A portion of the hierarchy from one such category is shown in Figure 11; the complete set of classes appears in Appendix B.

![Partial hierarchy for result classes](image-url)
The hierarchical organization of the sample, test, result, and outcome categories serves two purposes. First, it imposes a structure to make it easier for a designer to browse each category for an appropriate selection, that is, the hierarchies serve an indexical purpose. For example, a designer looking for a test to measure an object’s range and bearing could look under “dynamic test,” and then within that category, under “range and bearing test,” and from there could select “radar”. Second, the hierarchies are a means for propagating information from each category to its specializations, that is, they supply an inheritance mechanism. For example, tissue samples are defined to possess a “cell-shape” feature, so specializations of tissue sample, such as liver sample or blood sample, need not have this attribute explicitly encoded.

Expanding the knowledge base

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. Specifying a new class of outcome, for example, would involve identifying the superclass as well as the kind of results upon which this class of outcome is dependent. The example in Figure 12 shows a designer adding “temperature” to the dependencies (list of results) which are relevant to the new class “cause-of-change-in-ecosystem”.
Modifying existing classes works in much the same way. Note that IMap enforces constraints regarding class compatibilities, by accepting defined classes as the only legal fillers for class-to-class relations (e.g., the designer must select from existing result classes in building the results slot in Figure 12).

**Figure 12.** Defining a new test class

A Rationale for IMap

IMap answers the question “how can we help someone design an investigation” by providing an explicit investigation model, by making domain knowledge available, and by following a procedure for helping a designer instantiate the model. The central question is of course whether IMap makes the
designer’s task any easier. A secondary issue is whether IMap results in improved GBSs, without making the designer’s task any more difficult.

**IMap and the art of investigation**

Of main concern here is the extent to which IMap truly helps a designer create a GBS. An observation here is that people are more adept at imagining appropriate investigations than at conceiving of all the necessary details [4]. For example, a chemistry teacher may possess the requisite domain knowledge and a general design expressed 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. The usefulness of IMap thus lies in helping a designer articulate an idea as a well-specified investigation.

**IMap and the science of investigation**

The issue of IMap’s utility as a design aid is complementary to the question of IMap’s role in creating better GBSs. In other words, even if IMap did not make life any easier for a designer, it would still be advantageous to IDLE-Tool if it contributed toward better GBSs without making life any harder for the designer.
A rough measure of how an investigation could make its GBS better or worse is external coherency, which here means that the overall purpose of the investigation contributes to the user’s objectives in the scenario. In the absence of IMap, this question is never asked of the designer, since the conceptual information gained from the investigation has no direct interface analog and thus would not be considered by IDLE-Tool. IMap begins each design interaction with the question “what will your users be deciding?”, thereby establishing the investigation as something which ought to leave the user with some conclusive outcome. IMap goes further by supplying a library of the kinds of outcomes which are appropriate for this kind of GBS.

A second measure of an investigation is its internal coherency, which here means how well the different parts of the investigation fit together. Without IMap, it is up to the designer to supply, for example, a test which does in fact yield evidence relevant to the outcome the user is meant to determine. IMap contributes to internal coherency in two ways. First, it encourages the designer to keep in mind how pieces of the investigation are related, by prompting for design parameters in a way which emphasizes these relationships. For example, a designer of who specified an “index of refraction” result would be prompted for a test with “What yields an index of refraction result?” Second, IMap can determine, for any piece of the investigation template, which choices from its library would be consistent with those already installed in the investigation.

The internal and external coherencies of an investigation design are thus left to the designer in the absence of IMap, and are guided by an explicit model
with IMap. The point is that IMap can lead to a more effective GBS, by imposing constraints on the Investigation which, perhaps, force a designer to confront questions which had been left unanswered, but which enhance the contribution the investigation makes to the interaction.

Representational Tradeoffs

The sample-test-result-outcome representation admittedly excludes more complex notions of investigations, such as those involving multiple samples, iterations, and intermediate results. But the view of investigation adopted in IMap, while limiting, was abstracted from existing Investigate & Decide GBSs rather than being an *ad hoc* model adopted for convenience. And as Table 2 suggests, this basic model can support a reasonable range of potential applications.

**Table 2.** Completed GBSs within the scope of IMap’s knowledge base

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

Determining what, specifically, to include in IMap also relied in part on the GBSs which had been constructed using the tool. 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. At the same time, 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, 1968; BSCS, 1973a; BSCS, 1973b; BSCS, 1990; Edwards & Cimmino, 1975; Kroeber, Wolff & Weaver, 1965; Moore & Carlock, 1970), Chemistry (Ferguson, Schmuckler, Caro & Johnson, 1978; Garrett, Richardson & Montague, 1966; McGill, Bradbury, & Sigler, 1966; Metcalfe, Williams & Castka, 1982; Sutman, Harris & Greenstone, 1967; Vallarino, Quagliano & Kirkpatrick 1976; Weisbruch & Chewning, 1962), and Physics (Miner & Kelly, 1967; Stollberg & Hill, 1975; Taffel, Baumel & Landecker, 1970; Williams, Trinklein, Metcalfe & Lefler, 1973). These texts supplied information which shaped the sample, test, result, and
outcome class hierarchies. The representation of investigation knowledge in IMap was thus derived in part from empirical judgments and in part from a review of typical secondary science education activities.

**Extending the IMap: The next generation of IDLE-Tool**

IMap was constructed in order to address the absence of semantic and structural knowledge of investigations in the original version of IDLE-Tool. This basic notion of developing a richer knowledge representation and then bringing that knowledge to bear to assist the developer was preserved in the successor to IDLE-Tool, called Indie, a tool for authoring Investigate and Decide applications (Dobson, 1998). The extensions proposed below for IMap can thus also be viewed as design principles for related authoring environments (including Indie).

**Reusing investigation maps**

An important consideration is how we might adapt IMap to make previous investigations available to a designer, given that the task of creating an investigation is eased if a similar one were used as a starting point (Domeshek & Kolodner, 1991; Goel, 1989). IMap already stores the templates and scenarios which have been created for a particular GBS. We would simply augment the knowledge base with a collection of investigations (*i.e.*, templates and scenarios) from prior GBSs. A more challenging task is to allow IMap to retrieve the previous investigations most likely to be helpful to the designer. A straightforward
scheme would be to use the outcome as the index, so that retrieval then becomes a simple matching process, possibly weighting each retrieved investigation according to how many elements it shares with the investigation currently being defined. A slightly richer variation on this theme is to use IMap's (existing) class-subclass relations to derive a similarity metric based on graphical distance between nodes. A variant on this approach is to allow some level of user control over the matching preferences (Figure 13).

<table>
<thead>
<tr>
<th>sample</th>
<th>test</th>
<th>result</th>
<th>outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>concentration</td>
<td>cause of death</td>
</tr>
<tr>
<td></td>
<td></td>
<td>toxin in tissue</td>
<td></td>
</tr>
</tbody>
</table>

Find all investigations matching this description:

- animal tissue
- test class
- concentration in tissue
- cause of death

**Figure 13.** User-controlled match criteria for retrieving investigations

**Acquiring investigation knowledge**

The investigation libraries listed in Appendix B represent the object classes built into IMap. These libraries need to grow to accommodate a wider range of GBSs, and the information residing within these entries must be enhanced. A typical entry in the IMap library is designed to provide information that IMap can use to suggest compatible classes that could be used to create an internally consistent investigation. Although the current library possesses this data for only a few representative classes, IMap itself offers a simple mechanism
for partially automating the knowledge acquisition process. While creating an IMap, a designer can define a new class of object, and can assign that class to a specific location within the appropriate taxonomy. To further facilitate knowledge acquisition, the IMap could at this point ask the designer questions to elicit the features that are critical to the program’s ability to make suggestions. For example, if a designer were to create the class “dissolved oxygen test” as a subclass of concentration-test, the program could ask the designer to list the features which this test is capable of revealing.

The obvious expectation is that the IMap libraries will grow more robust the more the tool gets used. But this sort of growth must be leveraged judiciously, so that the integrity and singularity of the IMap libraries is preserved despite the contributions and modifications of multiple designers. This remains an open issue.

A better model of investigation

The sample-test-result-outcome model is sufficient for capturing the simplest investigations, but to support more realistic and interesting scenarios, IMap requires a deeper understanding of the structure of investigations. Four situations illustrate where a richer model would offer additional utility. First, results from one test can often influence a subsequent test, either by providing appropriate intermediate data, or by indicating which (if any) further tests to perform. Second, results from two tests may provide contradictory information. Third, determining a correct outcome may depend on multiple test results. Fourth, the results of a test are often simply to rule out, rather than confirm, an hypothesis.
To extend the IMap model so that the above aspects are accommodated, richer relationships among investigation elements are needed. For example, whereas an outcome in the current IMap is associated with one or more results, these associations would be elaborated to distinguish confirming from disconfirming results, or to indicate that a result may be necessary but not sufficient, or to point to a result that would rule out this outcome. An outcome could also be associated with competing outcomes. A consequence of this extended model of investigation is that the IMap would require additional heuristics for ensuring the integrity of an investigation: verifying the existence of at least one path to determining an outcome would rely on a grammar which included rules for each of the elaborated relationships mentioned above. For example, a rule might recursively encode that an outcome is reachable if (1) there are two results associated with it via “confirming” links, and (2) these links are AND’ed together, and (3) both results are reachable.

It may prove challenging to support the design of these richer investigations in the interview style of interaction employed in the current tool. This suggests that investigations instead be elicited in a less linear fashion. A possibility is to allow the designer to build a flow diagram of an investigation, with specialized connector nodes for representing more complex relationships like “supporting evidence”, “rules out”, and “necessary but not sufficient”. An example diagram appears in Figure 14 that depicts an investigation sequence for a GBS called Crime Lab Chemist.
Figure 14. Graphically specifying an Imap

Toward an intelligent design assistant

Given these extensions, the support IMap could lend a designer are illustrated in the hypothetical dialog depicted in Figure 15. In the dialog, IMap notices the absence of a conclusive sequence, suggests a test that closely matches the current IMap scenario, and applies a heuristic that less discriminating tests are generally run prior to more discriminating tests.
<table>
<thead>
<tr>
<th>Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>In your IMap, “poisoned” cannot be established by “presence of arsenic” alone. Should this be sufficient, or should we add another result?</td>
<td></td>
</tr>
<tr>
<td>ADD CONCENTRATION RESULT</td>
<td></td>
</tr>
<tr>
<td>What concentration result will help determine poisoned?</td>
<td></td>
</tr>
<tr>
<td>LETHAL DOSE</td>
<td></td>
</tr>
<tr>
<td>Are “presence of arsenic” and “lethal dose” sufficient for determining poisoned?</td>
<td>YES</td>
</tr>
<tr>
<td>How will the student determine lethal dose?</td>
<td>SUGGEST</td>
</tr>
<tr>
<td>I know of one investigation where chromatography yielded a concentration result to determine cause of death.</td>
<td></td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Will results other than “lethal dose” be suggested by any chromatography testing (specify)?</td>
<td>NON-LETHAL DOSE</td>
</tr>
<tr>
<td>What sample will show “non-lethal dose” when tested via chromatography?</td>
<td>LIVER SAMPLE 2</td>
</tr>
<tr>
<td>I notice that &quot;liver sample 1&quot; and &quot;liver sample 2&quot; yield the same result after immunoassay testing, but different results after chromatography testing. I am guessing that your students should perform the immunoassay test prior to the chromatography test. Should I add this ordering to the IMap?</td>
<td>YES</td>
</tr>
</tbody>
</table>

Figure 15. Hypothetical dialog with a more intelligent IMap

Evaluating IMap

The major piece missing from this puzzle is a systematic evaluation of IMap. The IDLE-Tool (and its IMap extension) is a prototype meant to demonstrate an approach to building case-based design tools for Goal-Based Scenarios. At least as important as conducting a formal evaluation of this particular implementation is developing findings about the underlying
methodology, because from this approach, a range of such design environments is likely to emerge. So one point is that, in considering evaluative data, it is important to distinguish between the authoring tool as an artifact and the tool as an instance of a general approach to design tools (Kass, 1998). The studies summarized at the beginning of this article (see also Bell, 1998) are a first step toward a more rigorous evaluation of the IDLE-Tool itself.

Two broad criteria that we can apply when evaluating a software design tool are its effectiveness as a design aid, and the overall quality of the applications that it helps produce. Although we cannot report findings that lend direct support to IMap’s effectiveness in supporting investigation design, subjects’ reactions to the original IDLE-Tool (regarding its emphasis on interface-level design and its lack of guidance at a conceptual level) were effectively remedied by the approach adopted in IMap (which was in fact created to address this problem). A teacher from the second study reported that “This (IMap) is nice. ‘What do you do with these samples?’ That’s what I had trouble with.” Another teacher from that study said that IMap “…cleared up some confusion I’d had about which were the samples and which were the tests.”

If we consider more generally a class of tools based on this approach, it should not be too surprising that users would respond well to this mode of interaction. Two observations support this expectation. First, the design process typically makes extensive use of past experience (e.g., Domeshek & Kolodner, 1991). Second, skilled designers tend to possess an understanding (i.e., an internal model) which helps guide the application of past experiences (Goel,
1989). Since the IMap architecture supplies both an explicit model and access to cases, it is likely that users will be comfortable with the way in which the tool proceeds through the design cycle. This brief analysis is not meant as a substitute for rigorous evaluation. Although much work is needed in evaluation, some data has begun to emerge from studies with the successor to IDLE-Tool, called Indie, that offers support for this approach as an effective way to create Investigate & Decide learning environments (Dobson & Riesbeck, 1998).

A second aspect of the tool to consider is the value of what it can help produce. Although evaluations of Sickle Cell Counselor (Bell, Bareiss & Beckwith, 1994) offer preliminary indications of the efficacy of applications produced with the tool (since the underlying model is the same), studies should be done to evaluate each new learning environment produced with the tool, so that, over time, an empirical picture of the tool’s educational worth can emerge. The Indie tool that succeeded the IMap/IDLE-Tool work has been used to develop dozens of applications (Dobson, 1998), several of which have been deployed and could be used to gather the data required for an evaluation of the principles implicit in the tool.

Finally, we can analyze the Investigate and Decide model with respect to some fixed set of principles for the design of learning environments. For example, Allan Collins has proposed six such principles (Collins, 1994); the model underlying IDLE-Tool is regarded favorably with respect to five of those six (Collins, 1995). Kass (1998) also offers some broad criteria for the construction
of such tools, that are based in part on the successes of the Investigate & Decide tools.

Conclusion

IMap is an extension to IDLE-Tool for supporting the creation of investigations. 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 is the obstacles typically faced in creating an investigation. The knowledge in IMap therefore comes from an analysis of the kinds of knowledge a designer applies to this process. This knowledge includes both a general component (a model of investigation), and specific knowledge about the elements within an investigation. IMap is useful as a design tool primarily because it enables designers possessing a general idea of the investigation they wish to create to define an explicit sequence of interactions, which are then operationalized as the Investigation in the GBS under construction. I have bounded the intended scope of IMap to support investigations appropriate for secondary science education, and have provided a rationale for why the knowledge built into the system is the right knowledge to include. Examples in the article have illustrated how such knowledge is brought to bear during interaction with a designer.
References


Bell


NOTES

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.

2. The three other GBS tools available at the time supported the design of evidence-based reporting, script-based, and production planning GBSs respectively.

3. Crime Lab was created by Smadar Kedar, Inna Mostovoy, Barbara Thorne, Mary Williamson, and the author.

4. Chris Riesbeck was instrumental in helping develop this argument.
Appendix A: Link Category Descriptions

This appendix lists the category description for the link types encoded in the knowledge representation. The reasoning engine currently uses the “yields”, “support”, and “proves” link types only.

YIELDS: Indicates that the named instance of SAMPLE, when analyzed by the named instance of TEST, yields the specified RESULT.

Ex: «sample: blood_1» & «test: mass_spec» YIELDS «result: arsenic_neg»

Note that a simulation is specifiable as a collection of YIELDS links.

SUPPORTS: Identifies the named RESULT as one which supports the validity of the named CONCLUSION.

Ex: «result: arsenic_pos» SUPPORTS «conclusion: arsenic_toxicity_pos»

WEAKENS: Inverse of SUPPORTS.

Ex: «result: arsenic_neg» WEAKENS «conclusion: arsenic_toxicity_pos»

PROVES: Similar to SUPPORTS, but stronger. Named RESULT is sufficient to prove named conclusion. If RESULT is a list, it means the conjunction of results proves the conclusion.
Ex: «result: arsenic_pos» (AND) «result: blood_toxic_pos» PROVES «conclusion: arsenic_toxicity»

DISPROVES: Inverse of PROVES.
Ex: «result: arsenic_neg» (AND) «result: blood_toxic_neg» DISPROVES «conclusion: arsenic_toxicity»

DERIVES: Not really a link instance, but an indication that the argument results will match an inference rule to produce the new result indicated.
Ex: «result: hg-type-aa» (AND) «result: hg-type-ss» DERIVES «result genetic-make-up-aa-ss»
Appendix B. Investigation library for IMap prototype

**sample-class**

- organic
  - animal
    - animal-tissue
      - regulatory-system
        - liver
    - excretory-system
      - fecal
      - urine
    - external-animal-tissue
      - hair
      - wing
      - teeth
      - claw
      - skin
      - fur
      - shell
      - nail
    - digestive-system
    - sensorimotor-system
    - circulatory-system
      - blood
    - reproductive-system
      - breast-tissue
    - skeletal
    - animal-individual
    - mammalian
      - marine-mammalian
        - seal
    - primate
      - human
    - plant
      - plant-individual
    - plant-tissue
    - fungus
    - simple-organism

- inorganic
  - synthetic
    - chemical
      - toxin
      - herbicide
    - textile
  - water
  - proximity
    - combustion
    - impact
    - flood
  - geological
    - fossil-fuel
    - mineral
      - rock
      - crystal
      - metal
    - air
    - soil
  - data
    - graphical-data
    - visual-data
    - image
    - video-data
    - symbolic-data
    - audio-data
    - textual-data
    - numeric-data
test-class

electrical-test
electrical-object-test
electroscope
electrical-circuit-test
voltmeter
ammeter
galvanometer
potentiometer
dry-cell
galvanoscope
electromagnetic-test
compass
magnet

physical-test
volume-test
gradiated-cylinder
volumeter
medicine-dropper
measuring-cup
beaker
density-test
spatial-test
meter-stick
tape-measure
micrometer-stage
micrometer
micrometer-ocular
ruler
mass-test
balance-triple-beam
balance-inertial
balance-analytical
balance-double-pan
scale
veterinary-scale
balance-spring
balance-platform
balance
visible-test
color-chart
x-ray
refractometer
lens-converging
lens-diverging
magnifying-glass
lens-hand
camera

thermal-test
temperature-test
calorimeter
thermograph-test
thermometer
temperature-change-test
refrigerator
matches
bunsen-burner
hot-plate
oven-drying
oven-incubating
autoclave
oven-sterilizing
chand
heat-of-reaction-test

calorimeter
thermograph-test
thermometer
temperature-change-test
refrigerator
matches
bunsen-burner
hot-plate
oven-drying
oven-incubating
autoclave
oven-sterilizing
chand
heat-of-reaction-test

chemical-test
chemical-composition-test
hemacytometer
hemometer
centrifuge
concentration-test
dissolved-oxygen-test
electrophoresis
geiger-counter
spectroscope
chromatograph
titrator
chromatograph-gas
filter-paper
assay-immuno
absorbtion-atomic
reactivity-test
ph-paper
litmus-paper
ph-meter
electrochemical-test
solubility-test
dynamic-test
range-and-bearing-test
soar-test
active-sonar-test
passive-sonar-test
radar-test
ceilingometer-test
laser-rangefinder-test
point-motion-test
stroboscope
duration-test
stopwatch
half-life-box
wave-motion-test
oscilloscope
<table>
<thead>
<tr>
<th>test-class (cont.)</th>
<th>result-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>biological-test</td>
<td>biological-property</td>
</tr>
<tr>
<td>physiological-test</td>
<td>physiological-property</td>
</tr>
<tr>
<td>pathological-test</td>
<td>pathological-property</td>
</tr>
<tr>
<td>mammogram</td>
<td>malignancy</td>
</tr>
<tr>
<td>biopsy</td>
<td>genetic-property</td>
</tr>
<tr>
<td>regulation-test</td>
<td>pattern-of-inheritance</td>
</tr>
<tr>
<td>ekg-test</td>
<td>phenotype</td>
</tr>
<tr>
<td>stethoscope</td>
<td>phylogenetic-property</td>
</tr>
<tr>
<td>reflotron-test</td>
<td>biological-distance</td>
</tr>
<tr>
<td>development-test</td>
<td>biological-classification</td>
</tr>
<tr>
<td>cellular-test</td>
<td>genetic-makeup</td>
</tr>
<tr>
<td>microscope</td>
<td>genotype-phenotype-link</td>
</tr>
<tr>
<td>microscope-compound</td>
<td>genotype</td>
</tr>
<tr>
<td>microscope-stereoscopic</td>
<td>sources-of-genetic-variation</td>
</tr>
<tr>
<td>genetic-test</td>
<td>regulation-property</td>
</tr>
<tr>
<td>reproduction-test</td>
<td>structure-of-endocrine-system</td>
</tr>
<tr>
<td>sexual-reproduction-test</td>
<td></td>
</tr>
<tr>
<td>asexual-reproduction-test</td>
<td></td>
</tr>
<tr>
<td>energy-production-test</td>
<td></td>
</tr>
<tr>
<td>respiration-test</td>
<td></td>
</tr>
<tr>
<td>behavioral-test</td>
<td></td>
</tr>
<tr>
<td>courtship-and-mating-test</td>
<td></td>
</tr>
<tr>
<td>social-test</td>
<td></td>
</tr>
<tr>
<td>locomotion-test</td>
<td></td>
</tr>
<tr>
<td>structural-test</td>
<td></td>
</tr>
<tr>
<td>environmental-test</td>
<td></td>
</tr>
<tr>
<td>response-to-stimulus-test</td>
<td></td>
</tr>
<tr>
<td>response-to-environment-test</td>
<td></td>
</tr>
<tr>
<td>test</td>
<td></td>
</tr>
<tr>
<td>response-to-substance-test</td>
<td></td>
</tr>
<tr>
<td>optical-test</td>
<td></td>
</tr>
<tr>
<td>light-source-test</td>
<td></td>
</tr>
<tr>
<td>photometer</td>
<td></td>
</tr>
<tr>
<td>diffraction-grating</td>
<td></td>
</tr>
<tr>
<td>light-meter</td>
<td></td>
</tr>
<tr>
<td>effect-of-light-test</td>
<td></td>
</tr>
<tr>
<td>lamp-ultraviolet</td>
<td></td>
</tr>
<tr>
<td>lamp-incandescent</td>
<td></td>
</tr>
<tr>
<td>effect-on-light-test</td>
<td></td>
</tr>
<tr>
<td>mechanical-test</td>
<td></td>
</tr>
<tr>
<td>force-test</td>
<td></td>
</tr>
<tr>
<td>barometer</td>
<td></td>
</tr>
<tr>
<td>stability-test</td>
<td></td>
</tr>
<tr>
<td>data-analysis-test</td>
<td></td>
</tr>
<tr>
<td>simulation-test</td>
<td></td>
</tr>
<tr>
<td>graphical-test</td>
<td></td>
</tr>
<tr>
<td>tabular-test</td>
<td></td>
</tr>
<tr>
<td>diagrammatic-test</td>
<td></td>
</tr>
<tr>
<td>result-class</td>
<td></td>
</tr>
<tr>
<td>biological-property</td>
<td></td>
</tr>
<tr>
<td>physiological-property</td>
<td></td>
</tr>
<tr>
<td>pathological-property</td>
<td></td>
</tr>
<tr>
<td>mammogram</td>
<td></td>
</tr>
<tr>
<td>biopsy</td>
<td></td>
</tr>
<tr>
<td>regulation-test</td>
<td></td>
</tr>
<tr>
<td>ekg-test</td>
<td></td>
</tr>
<tr>
<td>stethoscope</td>
<td></td>
</tr>
<tr>
<td>reflotron-test</td>
<td></td>
</tr>
<tr>
<td>development-test</td>
<td></td>
</tr>
<tr>
<td>cellular-test</td>
<td></td>
</tr>
<tr>
<td>microscope</td>
<td></td>
</tr>
<tr>
<td>microscope-compound</td>
<td></td>
</tr>
<tr>
<td>microscope-stereoscopic</td>
<td></td>
</tr>
<tr>
<td>genetic-test</td>
<td></td>
</tr>
<tr>
<td>reproduction-test</td>
<td></td>
</tr>
<tr>
<td>sexual-reproduction-test</td>
<td></td>
</tr>
<tr>
<td>asexual-reproduction-test</td>
<td></td>
</tr>
<tr>
<td>energy-production-test</td>
<td></td>
</tr>
<tr>
<td>respiration-test</td>
<td></td>
</tr>
<tr>
<td>behavioral-test</td>
<td></td>
</tr>
<tr>
<td>courtship-and-mating-test</td>
<td></td>
</tr>
<tr>
<td>social-test</td>
<td></td>
</tr>
<tr>
<td>locomotion-test</td>
<td></td>
</tr>
<tr>
<td>structural-test</td>
<td></td>
</tr>
<tr>
<td>environmental-test</td>
<td></td>
</tr>
<tr>
<td>response-to-stimulus-test</td>
<td></td>
</tr>
<tr>
<td>response-to-environment-test</td>
<td></td>
</tr>
<tr>
<td>test</td>
<td></td>
</tr>
<tr>
<td>response-to-substance-test</td>
<td></td>
</tr>
<tr>
<td>optical-test</td>
<td></td>
</tr>
<tr>
<td>light-source-test</td>
<td></td>
</tr>
<tr>
<td>photometer</td>
<td></td>
</tr>
<tr>
<td>diffraction-grating</td>
<td></td>
</tr>
<tr>
<td>light-meter</td>
<td></td>
</tr>
<tr>
<td>effect-of-light-test</td>
<td></td>
</tr>
<tr>
<td>lamp-ultraviolet</td>
<td></td>
</tr>
<tr>
<td>lamp-incandescent</td>
<td></td>
</tr>
<tr>
<td>effect-on-light-test</td>
<td></td>
</tr>
<tr>
<td>mechanical-test</td>
<td></td>
</tr>
<tr>
<td>force-test</td>
<td></td>
</tr>
<tr>
<td>barometer</td>
<td></td>
</tr>
<tr>
<td>stability-test</td>
<td></td>
</tr>
<tr>
<td>data-analysis-test</td>
<td></td>
</tr>
<tr>
<td>simulation-test</td>
<td></td>
</tr>
<tr>
<td>graphical-test</td>
<td></td>
</tr>
<tr>
<td>tabular-test</td>
<td></td>
</tr>
<tr>
<td>diagrammatic-test</td>
<td></td>
</tr>
</tbody>
</table>
result-class (cont.)

behavioral-property
  locomotion-property
  means-of-locomotion
  means-of-navigation
social-property
  courtship-behavior
  population-pattern
  population-growth-rate
  territoriality
reproductive-property
  means-of-reproduction
  parenting-behavior
environmental-property
  location-in-food-web
  dynamic-adaptation
tropic-response
thermal-property
  heat-of-reaction
  temperature
  boiling-point
  heat-of-crystallization
  heat-of-fusion
  specific-heat
  heat-of-vaporization
  response-to-heat
  freezing-point
  water-of-crystallization
  heat-capacity
  heat-of-combustion
mechanical-property
  mechanical-force
  coefficient-of-friction
  torque
  pressure
  mechanical-stability
  resonance-frequency
  center-of-mass
  moment
  center-of-gravity
dynamic-property
  property-of-motion
  acceleration
  wave-property
  wavelength
  waveform
  frequency
  amplitude
  range
  bearing
  rate-of-rotation
  velocity
  direction
inertia
meteorological-property
  dew-point
  sky-cover
  ceiling
  air-temperature
  precipitation
  air-pressure
  temporal-property
  duration
  half-life
electrical-property
  electrical-object-property
  electrical-charge
  impedance
  resistance
  electrical-polarity
  admittance
  capacitance
  conductance
electromagnetic-property
  inductance
  magnetic-flux
  magnetic-polarity
electrical-circuit-property
  voltage
  electrical-current
chemical-property
  electrochemical-property
  ionization-potential
  electronegativity
  molecular-polarity
  ionization-constant
reactivity
  activity-of-metal
  reaction-rate
  ph
  oxidation-state
  oxidation-potential
chemical-composition
  presence
  atomic-weight
  molecular-weight
  purity
  chemical-formula
concentration
  partial-pressure
  normality
  molarity
  molality
  moisture-content
solubility-property
  solubility-curve
  rate-of-solution
  solubility
  solubility-product-constant
<table>
<thead>
<tr>
<th>Physical Property</th>
<th>Outcome Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shape</td>
<td>Identification Outcome</td>
</tr>
<tr>
<td>Color</td>
<td>Artifact Type</td>
</tr>
<tr>
<td>Width</td>
<td>Artifact Behavior</td>
</tr>
<tr>
<td>Length</td>
<td>Motion Outcome</td>
</tr>
<tr>
<td>Height</td>
<td>Direction Outcome</td>
</tr>
<tr>
<td>Mass</td>
<td>Velocity Outcome</td>
</tr>
<tr>
<td>Volume</td>
<td>Artifacts Property</td>
</tr>
<tr>
<td>Compositional Analysis</td>
<td>Propagation</td>
</tr>
<tr>
<td>Correlative Analysis</td>
<td>Artifact Outcome</td>
</tr>
<tr>
<td>Dependencies Analysis</td>
<td>System Design Outcome</td>
</tr>
<tr>
<td>Variation</td>
<td>Biological System Design</td>
</tr>
<tr>
<td>Data-Trend</td>
<td>Pedigree</td>
</tr>
<tr>
<td>Causal Analysis</td>
<td>Causal Outcome</td>
</tr>
<tr>
<td>Behavioral Analysis</td>
<td>Cause of Event Outcome</td>
</tr>
<tr>
<td>Diffraction</td>
<td>Cause of Catastrophic Event</td>
</tr>
<tr>
<td>Reflectance</td>
<td>Cause of Fire</td>
</tr>
<tr>
<td>Spectral Composition</td>
<td>Cause of Flood</td>
</tr>
<tr>
<td>Polarization</td>
<td>Cause of Explosion</td>
</tr>
<tr>
<td>Index of Refraction</td>
<td>Cause of Physiological Event</td>
</tr>
<tr>
<td>Luminosity</td>
<td>Cause of Death</td>
</tr>
<tr>
<td>Response to Light</td>
<td>Cause of Change Outcome</td>
</tr>
<tr>
<td>Optical Property</td>
<td>Cause of Change in Artifact</td>
</tr>
<tr>
<td></td>
<td>Cause of Change in System</td>
</tr>
<tr>
<td></td>
<td>Cause of Change in Ecosystem</td>
</tr>
<tr>
<td></td>
<td>Cause of State Outcome</td>
</tr>
<tr>
<td></td>
<td>Optimal Choice Outcome</td>
</tr>
<tr>
<td></td>
<td>Optimal Treatment</td>
</tr>
<tr>
<td></td>
<td>Optimal Resource Allocation</td>
</tr>
<tr>
<td></td>
<td>Optimal Location</td>
</tr>
<tr>
<td></td>
<td>Optimal Route</td>
</tr>
<tr>
<td>Classification Outcome</td>
<td>Diagnosis Outcome</td>
</tr>
<tr>
<td></td>
<td>Predictive Outcome</td>
</tr>
<tr>
<td></td>
<td>Prognosis</td>
</tr>
<tr>
<td></td>
<td>Risk Analysis</td>
</tr>
<tr>
<td></td>
<td>Risk Assessment</td>
</tr>
<tr>
<td></td>
<td>Relative Probabilities</td>
</tr>
<tr>
<td></td>
<td>Contributing Outcome</td>
</tr>
</tbody>
</table>
Bell