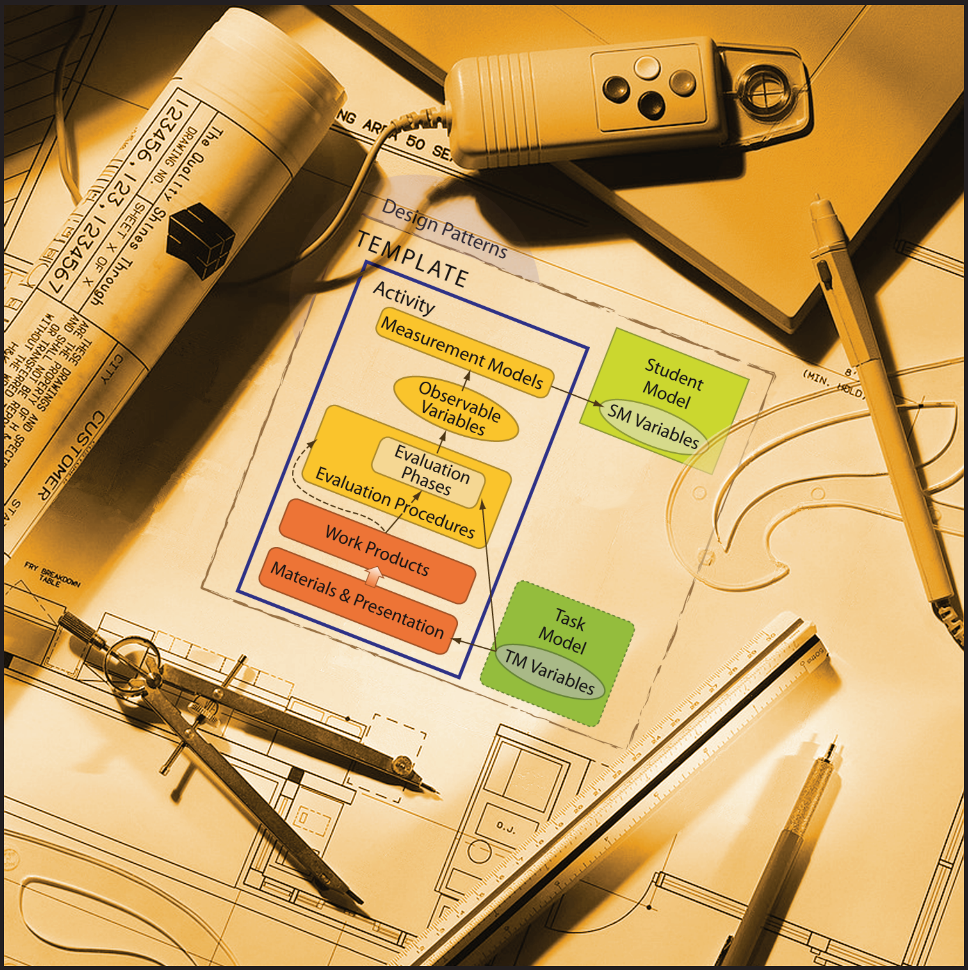

PADI Technical Report 21 (DRAFT)
March 2009

The PADI System as a Complex of Epistemic Forms and Games

PADI | Principled Assessment Designs for Inquiry

John Brecht, SRI International
Britte Cheng, SRI International
Robert J. Mislevy, University of Maryland
Geneva Haertel, SRI International
Kathleen Haynie, Kathleen Haynie Consulting

Report Series Published by SRI International

SRI International
Center for Technology in Learning
333 Ravenswood Avenue
Menlo Park, CA 94025-3493
650.859.2000
http://padi.sri.com

The PADI Design System
as a
Complex of Epistemic Forms and Games

John Brecht, SRI International

Britte H. Cheng, University of California, Berkeley

Robert Mislevy, University of Maryland

Geneva D. Haertel, SRI International

Kathleen A. Haynie, Kathleen Haynie Consulting

Paper presented at the
Annual Meeting of the American Educational Research Association,
April 8, 2006 San Francisco, CA

Abstract

We examine the Principled Assessment Designs in Inquiry (PADI) design system as a complex or system of epistemic forms with several epistemic games that may be played to complete the forms. This analysis illuminates what new understanding about assessment design is generated in the PADI process, thus illustrating the relative value of the PADI process to individual designers and to the field as a whole. PADI's forms are expressed in its underlying object model. The games are being developed by PADI's early users and represent a combination of standard practice in assessment design and new practice uniquely enabled by the PADI system.

1 Background – PADI

PADI is a computer-based design system that guides users through the multifaceted process of designing assessments, with a focus on assessments of scientific inquiry. PADI's products are not fully implemented assessments, but rather thorough "blueprints" based on which an assessment may be implemented. This approach provides the blueprint user with complete details of a coherent assessment argument and the steps required to realize that argument. These blueprints are referred to in PADI as Templates. The other primary products of the PADI design are Design Patterns. Whereas Templates provide the technical details of the operational elements of an assessment, Design Patterns provide the substance of an assessment argument. Design Patterns serve as a bridge between the expert's modeling of the domain of science inquiry and the specification of the technical details required for an evidence-centered assessment of science inquiry.

PADI's theoretical framework is Evidence Centered Design (ECD).[i] The primary tenet of ECD is that assessment design begins with evidentiary reasoning, not the design of tasks. Oftentimes assessment design is task-centric, wherein "tasks" are developed in isolation without collaboration among subject matter experts, assessment specialists, and psychometricians. Such tasks are often developed sequentially with each expert providing input and leaving it to the next specialist to determine how to score the task, render it, or determine what student proficiencies need to be included. In the past, it has been troublesome to see how often assessment tasks were designed without regard to the evidence needed to justify the assessment argument. On the one hand, this usual approach facilitates designing the simplest kind of assessments, namely those which are meant to measure an overall proficiency in a domain of independent tasks. On the other hand, the same approach falls short for designing coherent collections of tasks meant to measure multiple aspects of knowledge and skill, which may be required in different mixes in different tasks, and in which features of performance can be interactive or dependent on one another. When assessment design is evidence centered, the designers are led to consider the inferences they wish to make about students, the observations coupled with statistical models that can invoke evidence to inform those inferences, and finally the situations in which those observations can occur (i.e. tasks). The structure of the PADI templates further leads to an expression of a coherent assessment argument in the form of the more technical "machinery" of implemented assessments, such as task specifications, scoring procedures, and measurement models.

PADI's technological framework is Object Modeling. Object Modeling is a proven technique in software design (and more recently the design of business processes) for generating testable prototypes of software or other systems and processes. An Object Model represents a system or process as a set of "objects" each defined by an array of parameters and relationships to other objects. Such relationships include: containership, logical flow, inheritance, and workflow. PADI's object model schema has been designed to represent the necessary components of assessment design, consistent with principles of evidence centeredness. The PADI design system has been implemented as a web-based application, backed by a database to store users' designs, as well as a library of assessment components and Design Patterns.

2 Background – Epistemic Forms and Games (EF/G)

2.1 Collins and Ferguson's Definitions

In their seminal paper[ii], Collins and Ferguson (1993) introduce the concept of "epistemic forms and games". An epistemic form is some means of structuring information so as to guide inquiry, and an epistemic game is a process by which people gather, organize, and express information in terms of an epistemic form. Collins and Ferguson catalog a variety of epistemic forms and corresponding games that are used regularly by scientists to guide scientific inquiries and by historians to guide historical inquiries (Table 1). These two domains cover a wide variety of epistemic forms, but the catalog is by no means complete. In the process of creating this catalog, Collins and Ferguson also set out a framework for analyzing any potential epistemic form and game. In the course of this paper, we will use this framework to study inquiry and epistemic forms and games in the context of design and assessment.

The process of taking "raw" data (observations, measurements, records, etc…) and bringing them into such a structure is an epistemic game. For example, one can guide the inquiry "How many animals are there?" by going to some source of animal data (whether it be observations in nature, or simply opening a book about animals), and adding animal instances to the list. By applying constraints to this game, one can vary the epistemic nature of the list. For example, one could ask: "How many animals that live in the desert are mammals?" Now we are constrained to only add animals to our list the meet the qualifications of "lives in the desert" and "is a mammal." We could further qualify this game by constraining items on the list to be unique. Now we can guide inquiries about distinct species of animals, rather than absolute number of them.

Table 1 – Types of Epistemic Forms and Games

- Structural Analyses
    - Listmaking
    - Temporal decomposition
    - Compare and contrast (by decomposition of objects into lists of comparable properties)
    - Cost/Benefit analysis (Compare and contrast in economic)
    - Primitive elements game (describe a set of objects in terms of combinations of the same primitive elements that compose them all)
    - Cross product/table game (Two dimensional listmaking. E.g. a table of vehicles where one axis is medium and the other is form of propulsion)
    - Tree structure/hierarchy
    - Axiom systems (E.g. Euclid's geometry)
- Functional/causal Analyses
    - Critical event analysis (causes and consequences associated with an event)
    - Cause-and-effect (chain of causality)
    - Problem-centered analysis (study an event stream in terms of problems, solution, and problems that result from the solutions, and so on…)
    - Multi-causal analysis, AND/OR Graphs

    - Form-and-function analysis (break down system of objects in terms of their properties and the functions they perform)
- Process Analysis
    - System dynamics models
    - Aggregate behavior models (particle systems)
    - Constraint systems (Set of equations governs behavior of a system)
    - Situation-action models (characterized by a series of instructions like: “If situation is X, do Y.”
    - Trend and cyclical analysis (plot variable of interest over time and extrapolate function)

2.2 The Benefits of EF/G Analysis of PADI

Epistemic games are processes by which unstructured information is transformed into structured information that contains knowledge, the activity thereby guiding specific inquiries. The process is motivated by a query, though the query itself may ultimately be modified as the game progresses. Thus, epistemic game play, as a form of inquiry, is an iterative process. Collins and Ferguson sought to catalog important epistemic forms and games for the purposes of classroom instruction, as it is critical for science students to understand these fundamental tools that span all scientific inquiry. Beyond this immediate context and application, Collins and Ferguson’s (1993) article:

> “…outlines a prospectus for a theory of epistemic forms and games. We view it as a primitive-elements theory, in which we are trying to identify the primitive forms and games out of which theories in science and history are constructed. Systematic analyses of theories and inquiry strategies in the different disciplines are needed to build a detailed theory of the different epistemic forms and games cited here and to identify other forms and games that sophisticated inquirers use.” (p. 40)

The process that led to the development of PADI began with identifying the forms and games used in assessment design. A multitude of such forms and games can be found throughout the assessment literature (i.e., content/process matrix, task shells, test blueprint), each an aide to some process in designing or implementing an assessment of one kind or another. The forms vary considerably as to their structures and uses, but in no case is it explicit that all assessment is at its core, and in every detail in use and construction, an exercise in evidentiary reasoning. PADI has sought to bring this fundamental fact to the fore, in terms of explicit structures that unify previous forms and capture the essential elements of all assessment design (although see Messick, 1989, 1994, Kane, 1992, and Wiley, 1991, on validity from the perspective of evidentiary argument). In this paper, we walk through the two primary epistemic forms of PADI – Design Patterns and Templates – and we investigate the games that are played to complete these forms and what it is that makes these games epistemic. In doing so, we demonstrate that the PADI system is not merely an attempt to establish a standard or the creation of a formal structure for technological convenience, but rather that the PADI design process actually generates new knowledge. This new knowledge arises primarily through the design principles that PADI encourages and enforces. For example, principles of object oriented design can encourage designers to think of a particular assessment as an instance of a broader class of assessment which can be generative of designs for other specific assessments, as well as the general framework spanning those assessments.

By considering the PADI design process as an epistemic game, we illuminate the fact that assessment design is a process of discovery; that one cannot simply arrive at the "answer" through pure contemplation, but rather by carrying out a principled procedure. In so doing, we can analyze this procedure and perhaps optimize it or at least provide guidance as to its best application. Further, just as PADI's framework will be shown to be useful for creating understanding of how a given assessment fits into a broader landscape of related assessments, we expect that the Epistemic Forms & Games framework will illustrate what other games are in the neighborhood of PADI. Specifically, we may come to understand just what kind of design tool PADI is and what related purposes beyond assessment it may be suitable for.

3 PADI's Epistemic Forms and Games

3.1 Inquiries Guided by PADI

The most critical component in the description of an epistemic form and the related games is the type of inquiry they can guide. In PADI, users engage in an inquiry that explores how to manifest their assessment goals. Briefly, Design Patterns lay out potential design choices within the elements of a coherent assessment argument, at a narrative level; Templates are specifications for the machinery that implements an assessment argument, that is, technical information about activity flow, evaluation procedures, stimulus materials, measurement models, and the like.

Using the epistemic form of a Design Pattern, users begin by identifying aspects of students' knowledge, skills or abilities (KSAs in PADI terms) that they wish to measure, as described for example in an educational standard, or in terms of capability to carry out certain kinds of work in certain situations.

A user of the PADI design system creating an assessment template may begin with a relatively unstructured conceptual assessment argument, generally in terms of the elements or processes of particular, familiar, kinds of assessments. For example, they may have a notion of the KSAs they want to measure, the observations they want to collect, or features of the tasks the user think may be relevant. Users seek to determine what assessment can optimally elicit evidence about students' knowledge, skill, or ability in some area.

The completed epistemic form that the PADI game guides the user to is a Template, a blueprint specifying the components of the operational assessment and their interrelationships.

3.2 The Epistemic 'Forms' of PADI: The PADI Object Model

The PADI design system was intended to guide inquiries regarding appropriate evidence-driven assessment designs for particular inquiry skills. Assessment designs are multifaceted both structurally and functionally, and thus call for the thoroughness of object modeling. Likewise, as a design system, PADI must produce blueprints sufficient for implementers to design assessments with minimal further guidance. Object models are ideal for providing such blueprints, given their track record in the software industry and business for producing testable prototypes of complex programs and processes.

We first consider the initial phase of object modeling – the creation of a schema for the domain. PADI's schema is motivated by Mislevy, Steinberg, and Almond's previous work on a Conceptual Assessment Framework[1]. The PADI schema is presented visually in Figure 1.

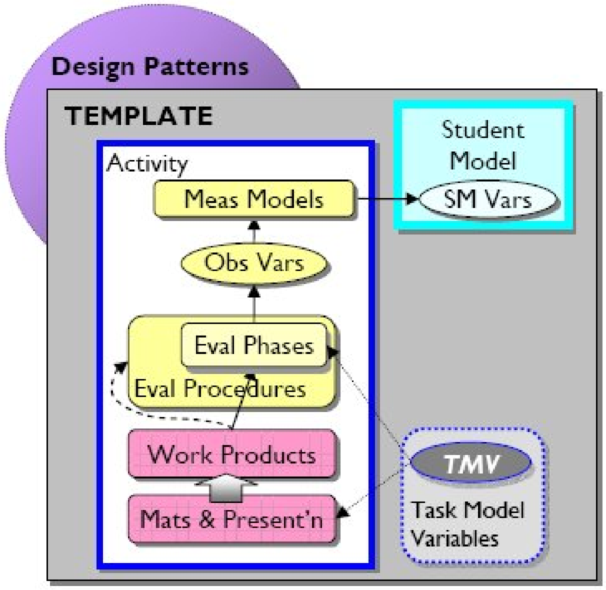

Figure 1 – The PADI Object Model Schema

As mentioned, the two primary products of the PADI Design System are Design Patterns and Templates. Figure 1 shows Design Patterns sitting in the background of a Template as they may inspire Template design, but they are not directly a part of a Template. Design Patterns are described almost entirely by parameters, with little internal structure. The only objects Design Patterns link to are other Design Patterns, either through subcomponent relationships ("part of") or inheritance ("kind of") relationships.

Templates on the other hand have significant internal structure. A Template describes one or more Activities. Activities are comprised of a series of components related to each other in a linear logical flow and workflow starting with the initial Materials and Presentation supplied to the students (such as specifications of the setting, instructions, tools, and so on, including, for example, specifications in a simulation environment). Students performing the task then produce Work Products (from item responses, to solution paths, to plans and execution of science investigations). Work products are then passed to Evaluation Procedures, which may contain one or more internal Evaluation Phases. Evaluation procedures produce Observable Variables ("scores"), which can be passed to Measurement Models. Measurement Models use statistical techniques to relate Observable Variables to Student Model Variables.

The superset of Student Model Variables of all the Template's Activities is contained by the overall Student Model for the Template. The Student Model is an attempt to model internal Knowledge, Skills, or Abilities with respect to a particular domain. A Template can also contain settings of Task Model variables in the Template's Activities' Evaluation and Materials & Presentation settings, e.g. – to set a degree of scaffolding. Task Model variables represent ways in which the assessment task can be varied, for example, number of steps required in a problem solution (Mislevy, Steinberg, & Almond, 2002).

In addition to internal structure, a Template may also be part of an external structure of other Templates primarily through inheritance relationships, but sometimes as actual subcomponents of large connected task structures. A distinction is made in PADI between abstract and concrete Templates. An abstract Template leaves some of its parameters or even entire subcomponents undefined or vaguely defined. "Child" templates may then extend the Template by supplying more specific information. In this way, general Templates may be defined that form the root of whole families of related assessments. The "family tree" of inheritance may be arbitrarily deep. A completely defined Template is called "concrete" and is referred to as a Task Specification. Task Specifications are the products from which implementers may generate working assessments.

3.3 The Epistemic Game of PADI: The Design Process

The second phase of Object Modeling is creating instances based on the schema generated in the first phase. The epistemic game played by users of the PADI system, the development of a Conceptual Assessment Framework (or CAF), is manifest as they work within PADI to create assessment tasks or families of tasks. Entry conditions in the PADI design game relate directly to the inquiry being pursued. A user seeking to express the substance of an assessment argument or a general practice that could apply to many variations of tasks on the same themes is in position to pursue a Design Pattern. If a user knows what knowledge or skills they want to measure and the evidence that needs to be collected and the type of task that will best elicit such observations, then they may pursue a Task Template.

As Design Patterns are typically unitary, or at most, part of homogenous structures that only contain other Design Patterns, the game for generating one is therefore less complex (though not necessarily less difficult) than the Template design game. All decisions about constituent elements (as general forms) have effectively been made ahead of time in the schema design. The user's task then is to populate those elements in their instantiation. The creation of a Design Pattern is more a subjective task than a mechanical one. The designer's goal is to capture the substance of an assessment argument such that it may be useful in a wide variety of situations. The Design Pattern should be *transparent*, meaning its name and summary should quickly indicate to users of the Design Pattern library what the Pattern is for, and whether they can use it. *Generality* is important; the Pattern should be no more specific than absolutely necessary to convey its essential argument.

Transfer may occur in the Design Pattern game if and when a designer realizes that the Pattern they are creating can be used to guide the creation of a Template. This primarily arises when the designer realizes that what they are creating depends on some specific assessment component, such as a particular Student Model or Activity. Another form of transfer can occur if the designer realizes that the Design Pattern they are creating is in fact a subset of another Design Pattern, or itself should be decomposed into a set of sub-patterns. Similarly, if the designer realizes their Design Pattern is a special case of some other Design Pattern, or vice-versa, then an inheritance hierarchy must be considered. In either case, the designer does not transfer entirely out of the Design Pattern game; they simply transfer to a form that spans a complex of Design Patterns.

The Template Design game may be simpler conceptually, but it is much more complex structurally. Templates have a multitude of components to be pieced together to create a sound, evidence-centered assessment design. The game begins with the designer's conceptual assessment argument. The designer must consider whether they are creating:

- A single complete Task Specification,
- An abstract Template for others to build upon,
- A family of Task Specifications based upon one Template, or
- A hierarchy of abstract Templates and concrete Task Specifications

In virtually any of the above cases, the designer does not necessarily start from scratch. They should first examine existing work to see if there are Templates or Template hierarchies that can be reused or added to. Thus, the overall entry conditions for the Template creation game are a combination of the scope the designer intends to work at and what existing work can be leveraged in the process. From here, many paths are possible. Users may start with a bottom-up approach, designing one concrete Task Specification completely, then abstracting "upward" from it to generate Templates from which other "sibling" Task Specifications may be generated. User may also pursue a top-down approach, laying a foundation for their family of tasks with the creation of a highly abstract Template capturing the core features of their family of tasks, then working "downward" creating more concrete Templates based upon the original abstract root and ultimately the family of fully concrete Task Specifications.

Within a given Template, a number of paths are also possible. In principle, designers are free to create their Student Model, Task Model, and Activities (as well as their subcomponents) in any order they prefer. Ideally, in keeping with principles of Evidence-Centered Design, designers create a Student Model first, then proceed to consider Activities that can elicit evidence for that model, and so forth. It is nonetheless possible for a designer to begin with an Activity, and work their way "out". However it is not possible to completely design an Activity without eventually considering evidence, as evidentiary reasoning is enmeshed in the Activity object. An Activity cannot be completely designed without specifying Evaluation Procedures, Observable Variables, and a Measurement Model. The Measurement Model cannot be complete without ultimately being linked to a Student Model. Thus, the PADI Template Schema ensures that all designers must consider their evidentiary assessment argument. The interconnections among the elements of the abstract Template structure ensure that the implemented elements of a particular task designed from it will be coherent.

4 Investigations of Emerging Practice

Having considered the PADI game theoretically, we now move on to examine a few case studies of PADI's early users. While the preceding discussion of the PADI design process referred to a single "designer" using the system, most of the significant PADI projects so far have been performed by teams of designers, with members representing various expertise such as psychometrics or content knowledge. Working in teams introduces other dynamics to the PADI game, such as collaborative editing of the target form and application of specific expertise in the conceptualization and design process. In practice, most design efforts began not merely with a conceptual assessment argument, but in many cases a complete, existing implemented assessment. In these cases, users were either looking to improve upon the existing tasks, or expand on it to create other similar task designs.

Four design teams were studied, all early users of the PADI system. In three cases (FOSS, GLOBE, and BioKIDS) designers were sent individual written surveys, which were followed up by group interviews. In the fourth case (Floating Pencil), the design process was observed directly over the course of four months.

4.1 FOSS

The Full Option Science System (FOSS) is an NSF-funded, inquiry-based curriculum that was selected as an implementation site in the PADI Project. (See http://foss.org/) This well-established, hands-on curriculum, under the direction of Dr. Linda Delucchi, Dr. Larry Malone, and Dr. Kathy Long from the Lawrence Hall of Science at the University of California, Berkeley, was selected as a PADI partner for its breadth of science coverage and wide implementation. With middle school curriculum units available on most key science topics typically covered at the middle school level (i.e., Electronics, Population and Ecosystems, Force and Motion, Diversity of Life, Planetary Science, Earth History, Human Brain and Senses, Weather and Water), these curriculum units contained not only science content and inquiry content and instructional activities, but both formative and summative assessments, as well. FOSS is one of the most widely implemented inquiry curricula in the nation and has sites that include schools with a range of demographic characteristics, teachers with varying training and experience as science instructors, and students that represent a wide variety of cultural and economic backgrounds.

The FOSS curriculum developers' have faced the challenge of designing an assessment system. The assessment system that they had designed in the past was formative and summative in purposes, addressed a wide span of grade levels, and measured both science content and inquiry skills. Thus, with regard to the PADI Project, FOSS represents a mature inquiry-based curriculum with implementation sites that include teachers and students of varying characteristics which would be helpful in understanding the conditions and practices that best support the use of the PADI assessment design system. In terms of assessment, the FOSS Project had developed embedded assessments as part of their curriculum and had a framework of science topics and progress levels that had influenced the development of their assessments.

As part of the PADI Project, FOSS' first design effort was to create a ten problem test focusing on speed, distance, and acceleration that was based on a previous conceptual design. The FOSS team initially set out to design ten Templates, one for each problem on the test. They quickly realized as they developed the Templates that there was a great deal of overlap across items. As a result, they chose to transfer them to one Template containing ten activities. Thus this group "discovered" on their own some of the value of Object-Oriented design principles, particularly reusability, which will be discussed further below. In that theme, they also realized that certain subcomponents of their activities, such as Observable Variables, had been individually created for each Activity, but were in fact, all instances of the same underlying Observable Variable. They were then able to drastically reduce the complexity of their design. From the interview:

> *"Originally these all said, "Equation Choice for Problem 1", "Equation Choice for Problem 2", and finally we sort of realized that they could be simplified, that we didn't have to preserve the intermediate observable variables, but just the final ones, as unique instances that the scoring engine would need."*

FOSS's next design effort was to create a family of tasks collectively known as the "ASK Assessment of Science Inquiry". The designers began with a "sketched out structure of what kind of aspects of inquiry we wanted to look at." From there, they considered existing Design Patterns to guide their work, but ultimately chose to create their own new Pattern, which was dubbed: "ASK Principles of Scientific Inquiry." This Design Pattern contained four aspects of inquiry as its Focal Knowledge, Skills, and Abilities parameter. The team then began design work on an abstract Supertemplate to act as the root of their task family tree, and then a more concrete Template. The designers reported frequently going back and forth between outside design elements in the form of papers and diagrams and the PADI Design System. They likewise frequently iterated within the Design System on their initial Design Pattern, using it to record in narrative form the high-level ideas that emerged from their more technical work in the Template design process.

Team members reported that the design system was a useful place to record their thinking, particularly their assessment argument, and their rationale for choosing particular items and that it was a very useful method for tracking the wide range of details that arise in assessment design. They also reported that it was helpful to be able to look at other Design Patterns to see other potential observable variables.

4.2 GLOBE

GLOBE is a "worldwide hands-on, primary and secondary school-based education and science program" (http:www.globe.gov.). Students in GLOBE science classes take measurements related to earth science content investigation areas, such as atmosphere, hydrology, and land cover. These data are archived, posted on the Web, and can be used for research purposes by students and teachers who participate in the GLOBE program. While working with the GLOBE program, SRI International developed several assessment resources, such as a template for developing classroom assessments and a generic rubric, to support teachers in assessing their students' understanding of scientific inquiry using GLOBE data. In the PADI Project, the GLOBE curriculum was selected because of its conceptualization of inquiry. In GLOBE, inquiry is defined as a cycle of phases and not as discrete, unrelated activities (as per Stewart & Hafner, 1994, and White & Frederiksen, 2000).

The GLOBE team began their work with a template of sorts already, which had been designed under a previous (completed) project. The previous work had encouraged this team to think about the components of their assessment. The team began by combing through existing GLOBE assessment work, to determine the set of skills they were interested in assessing. They chose a set of existing GLOBE activities that would test those skills to be represented in PADI. The team reported that the transition to PADI was fairly natural, and that the assessment components already were familiar to them from their previous work, but that the PADI structure made them think more deeply and thoroughly about the assessment components and their linkage.

When a new team member joined that was unfamiliar with the PADI components, he began to make visualizations to aid his reasoning and communicate ideas. These visualizations were ultimately able to assist all team members in understanding logical flow and also became artifacts that mitigated change—tracking the evolution of the Template in a way that the existing PADI implementation does not. The visualizations helped in error correction:

> *"So at the beginning our connection among our task model variables are so complex, and I couldn't make a good figure. That is when I realized there is something wrong with our task model variable setting or development."*

At first the team planned on creating separate Templates for each grade level, but then realized that they could make just one Template specifying as a Task Model Variable a range of possible grade levels. The particular grade level setting is designated when the Template is instantiated as a Task Specification. Still, there were other dimensions along which the team wanted their activities to potentially vary, far too many for a simple hierarchy:

> *"It is almost more useful sometimes to think of like a modeling tool, where you are setting multiple sliders. You know, so that for each question you are asking someone to answer about complexity or scaffolding or inquiry skills, all of those sort of Template level questions, they are almost like sliders on a modeling tool. But you have got so many variables going at once, that is why it is hard to think of a… I mean even a tree structure would be a tree with many, many branches."*

This led the team to pursue a "Wizard" approach, both for guiding their own creation of new Templates and to allow outside users of their Templates to more easily find what they need. The Wizard is essentially an automated interview that poses a series of questions to a user, and based on the users' responses, guides them to a Template (or perhaps dynamically generates a Template based on an abstract Supertemplate in a future implementation.) The primary branching in their Wizard tree is based on the Student Model the user wants to instantiate. The team wanted to support four possible models:

- Various inquiry skills represented separately (multiple inquiry), various kinds of content knowledge represented separately (multiple content)

- Various inquiry skills represented separately (multiple inquiry), various kinds of content knowledge combined as a single value (single content)
- Combined inquiry and content (in one Student Model Variable)
- Inquiry skills combined as a single value (single inquiry), multiple content

The team reported the following benefits of using the PADI process:

> *"For me it is making the implicit explicit and thinking about all the complexities involved in developing a GLOBE Assessment"*

"It isn't just in your head, in the head of one or two other staff people who are working on a project dealing with design. But because you have made so many of these features and decisions about the features more explicit, you have more of an opportunity to move into the kind of rapid design and prototyping that PADI supports."

4.3 BioKIDS

The BioKIDS team seeks to determine what assessments can provide information about students' knowledge, skills, and abilities (KSAs) with respect to the BioKIDS curriculum. These KSA's include both content knowledge and inquiry reasoning skills. In particular, the sequence of sixth grade curricular units focuses on the fields of biodiversity, weather, and simple machines. Within each of these content areas, three inquiry skills are carefully scaffolded and fostered: creating scientific explanations, interpreting data, and making hypotheses and predictions. With our sixth grade students, the BioKIDS curricular sequence is often the first time that they are exposed to inquiry-based science. Therefore, it is imperative that the assessment system that we use focus on having assessment items that assess content and inquiry skills at different levels of difficulty.

BioKIDS: Kids' Inquiry of Diverse Species, is another of the curriculum partners selected for participation in the PADI Project. (See http://biokids.umich.edu). Under the direction of Dr. Nancy Butler Songer, this middle school focused curriculum has developed inquiry-based curriculum units in the topical areas of ecology, weather, and simple machines. NSF-funded, this innovative curriculum has been implemented in many classrooms in the Detroit Public Schools over the past four years. The BioKIDS curriculum was selected as a partner to the PADI Project for several reasons, including its relatively recent development as an inquiry-based curriculum that had incorporated some technology-based instructional supports. In addition, the scaling of BioKIDS was to occur in an urban school setting which would provide use of the assessment design system in educational settings that are charged with educating large numbers of underserved students. While the BioKIDS developers had created some new assessment tasks to be administered as part of their curriculum units and had identified some assessment tasks from existing science examinations, they had not established a formal assessment system to support their entire curriculum.

The BioKIDS team began with a core set of inquiry skills they wished to foster systematically and sequentially throughout three consecutive curricular units for sixth grade science. They then proceeded to comb through the library of Design Patterns for existing useable patterns. Ultimately, the team found one Design Pattern that fit their needs (after a little refinement), and went on to generate two more Patterns from scratch.

(The team was using the Design System in the early stages of the PADI project, when the library of Design Patterns was still under development.)

The team then proceeded to develop their Templates. BioKIDS used yet another pattern for structuring their Templates – a conceptual structure called the Content-Inquiry Matrix--that - served as a mechanism for mapping task complexity along two dimensions (content and inquiry). Three Content-Inquiry Matrices were developed, one for each Design Pattern/Area of Inquiry Reasoning (e.g. building scientific explanations from evidence, analyzing data, and constructing hypotheses and predictions). A sample Content-Inquiry Matrix associated with Building Scientific Explanations from Evidence is presented in Table 1.

The team worked on their corresponding set of Templates iteratively. For example, they left Measurement Models for all Templates for last, as they did not have the expertise on hand at the beginning of their effort. After consulting with experts on the missing content, the team revisited and refined their Templates until they were complete and worked well with both of the other conceptual structures (design patterns and Content-Inquiry Matrices).

Table 1: Levels of Content and Inquiry Knowledge Needed for BioKIDS Assessment Items Related to the Design Pattern: "Formulating scientific explanation from evidence"

	Simple minimal or no extra content knowledge is required and evidence does not require interpretation	**Moderate** students must either interpret evidence or apply additional (not given) content knowledge	**Complex** students must apply extra content knowledge and interpret evidence
Step 1 Students match relevant evidence to a given claim	Students are given all of the evidence and the claim. Minimal or no extra content knowledge is required	Students are given all of the evidence and the claim. However, to choose the match the evidence to the claim, they must either interpret the evidence or apply extra content knowledge	Students are given evidence and a claim, however, in order to match the evidence to the claim, they must interpret the data to apply additional content knowledge
Step 2 Students choose a relevant claim and construct a simple explanation based on given evidence (construction is scaffolded)	Students are given evidence, to choose the claim and construct the explanation, minimal or no additional knowledge or interpretation of evidence is required	Students are given evidence, but to choose a claim and construct the explanation, they must interpret the evidence and/or apply additional content knowledge	Students are given evidence, but to choose a claim and construct the explanation, they must interpret the evidence and apply additional content knowledge.
Step 3 Students construct a claim and explanation that justifies claim using relevant evidence (unscaffolded)	Students must construct a claim and explanation however, they need to bring minimal or no additional content knowledge to the task	Students must construct a claim and explanation that requires either interpretation or content knowledge	Students must construct a claim and explanation that requires the students to interpret evidence and apply additional content knowledge.

4.4 Floating Pencil

Large scale reference science exams like NAEP are very important in education to inform the public of student achievement levels and inform educational policy decisions. Traditionally these assessments have not focused on performance assessments of scientific inquiry; they have provided, primarily, information about scientific content. Furthermore, NAEP performance assessments are often inquiry-constrained and content-lean[iii,iv], a design that does not represent the full range of possibilities for performance assessments. Through the lens of the evidence-centered design framework, the Floating

Pencil team seeks to determine the characteristics of the Floating Pencil assessment, what the assessment is capable of measuring, and what a family of similar performance assessment tasks might look like.

The Floating Pencil team sought to "reverse engineer" an existing middle school performance assessment in science inquiry. This collaborative team consisted of seven members: a psychometrician, four educational psychologists, an assessment designer, and an engineer. Their intention was to understand how items from a large-scale reference exam would be represented in a PADI Template or Task Specification. An available pool of NAEP and other items from large-scale examinations was analyzed in terms of usefulness for measuring inquiry, the NSES standards, and natural item groupings or theme blocks. The team selected the NAEP Floating Pencil task, a set of 14 items of various types (multiple-choice, short answer, short essay). The task prompts students to conduct a hands-on investigation in which the research question has been posed and procedures have been specified. Students are asked to carry out experimental procedures following standard methods so that comparisons can be made, data summarized, predictions made, and explanations provided.

Templates and Design Patterns were initially used in an exploratory fashion by the Floating Pencil team. In doing this, the team's understandings and definitions of Student Model, Evidence Model, and Task Model were iterative. A partial Template was drafted. In doing this, the Floating Pencil task was decomposed according to PADI's structure and objects; the team distinguished Work Products, Materials & Presentation, and Observable Variables. Activities were defined as unique scorable Work Products. In this way, item-level information was combined for Measurement Models, Evidence Models, Materials & Presentation, and Presentation Logic. The group discussed whether to have 1 Template with 14 Activities, 1 per item; 14 Templates, 1 per item; or 1 Template with 10 Activities defined on the basis of the Observable Variables. It was decided to pursue 1 Template with 10 Activities. The reason for this was the conditional dependency of all the Floating Pencil items on a common task directive and physical stimulus. Much discussion was given to handling the conditional dependencies of the items: pairs of sequentially dependent items, and all items sharing a common physical stimulus. The team then drafted a Design Pattern. This prompted thinking about the Student Model and Evidence Model; also, the relationships between Task Model Variables and quality of measurement were explored. Two roles of Task Model variables were also discussed extensively – how they contribute a source of error variance and how they can contribute to the "true score" or "construct" being measured. Finally, the Floating Pencil Design Pattern was compared with a pre-existing Design Pattern and determined to be a special case or instance of that pre-existing pattern.

To facilitate the group's learning curve with the PADI Design System, a bottom-up approach to PADI was undertaken – the reverse engineering of the Floating Pencil task into a Task Specification. In doing this, the group necessarily grappled with the specifics of Measurement Models and Student Models. It was decided to mirror, as closely as possible, the cognitive framework of the test developer, NAEP; this framework is a 3 X 3 content by process matrix. Student Model Variables were defined on the basis of this framework. Task Model Variables and Materials and Presentations were distinguished at the Task Specification and Activity levels. The group gave much discussion to the definition of Task Model Variables, resulting in the goal of defining a ‘Family of Tasks’ for Floating Pencil.

The Floating Pencil team abstracted upwards to generalized PADI forms representing a ‘family of tasks’. The guiding questions for the team’s consideration of a task family can be stated as “The Floating Pencil task is an instance of what?” and “How will a Floating Pencil task family be represented in the PADI System?” The team considered generating a Template or group of Templates on the basis of task scaffolding levels, what a generalized inquiry ‘family’ of tasks might look like, and the characteristics of interchangeable versions of the task. Different bases for the task family was explored including scaffolding levels, a set of already existing constructs, and task characteristics of already existing performance assessment. One possible family structure was based on the segments of the inquiry cycle. For this purpose, the scientific inquiry process was these phases. Then when an assessment of inquiry is created, these Templates can serve as “building blocks.” Designers may assemble an assessment using those subcomponents relevant to the phases of inquiry they wish to assess, embellishing with requisite material for coordinating work across phases.

5. Object Modeling

Object Modeling is the technological framework of PADI. While the previous sections have focused on the conceptual and substantive aspects of the PADI design system, this section discusses the underlying object model in somewhat greater detail. Object models are themselves epistemic forms, at a greater level of generality than the PADI implementation, with a vast and growing collection of examples, tools, and processes that have been developed in a variety of fields for using them to design and analyze business processes and information systems (computer software, in particular). Many of the properties of the PADI design system as an epistemic form, and the epistemic games that are evolving to use it, can be viewed as a legacy of its object-modeling foundations.

The components of the Conceptual Assessment Framework[1] have been expressed in PADI as an object model schema[v] in the Unified Modeling Language (UML)[vi]. The principles underlying Object Modeling are the principles of Object-oriented Design. Object-oriented design originated in object-oriented programming languages such as SIMULA-67[vii] and Alan Kay's well-known Smalltalk[viii]. Object-oriented programming treats computer programs as systems of "objects" (originally thought of as literal physical objects) that interact through message passing. Inner workings and states of objects are concealed (*encapsulation*); functionality is exposed through an "interface", essentially a well-defined set of inputs and outputs. The purpose of encapsulation is to prevent other objects from altering the state of an object in potentially damaging ways. It also leads to "clean" separation of functionality into components like physical objects that are thus reusable. *Reusability* is a critical principle of object-oriented programming. Another important principle is that of *inheritance*. An object may be defined as a child or subclass of another class of object, in which case it *inherits* that object's properties and *extends* them with its own properties. For example, a convertible car inherits all the properties of "car" and extends them with the property of a roof that can fold back. In the PADI design system, all of the features of object-oriented design are leveraged to facilitate evidence-centered assessment design. For example, once users have articulated student model(s) that describe the ways that KSAs should be expressed, these pieces of the underlying argument could be drawn into the design process of a related assessment.

In modeling, one seeks to create an approximation of some complex system or object (or system of objects) that provides a compact, transparent, and potentially functional representation of the system, salient for some context. Compactness allows for a degree of portability of the model, as well as allowing it to be more immediately understandable. (We can "wrap our head around" the compact model more easily than the full featured real thing.) This portability and understandability are key features to the model's function, which is communication. In order to be compact, we must know what details can be stripped away and what cannot; we must know what is salient. Determining *saliency* is the art of modeling. A physicist seeking to represent the kinematic properties of a ball can work from a theoretical abstraction that is only concerned with the ball's mass, geometry, and coefficient of restitution, throwing away details such as the ball's color, whereas a child picking a toy ball off a shelf at the store may *only* concern herself with the ball's color.

Models can be physical abstractions, such as an architect's miniature model of a building constructed with cardboard and plastic; mathematical abstractions, such as the physicist's model of a ball mentioned above; or something in between, such as a CAD model of a machine part. Object modeling is a conceptual abstraction that may contain some mathematical elements, but largely contains textual descriptions of components of a system, a web of relationships connecting them, and optionally, procedures ("methods") an object may have to generate, transform, or convey information. In addition to the general principles of modeling, object modeling is further guided by principles of object-oriented programming

The original purpose of object modeling was to create representations of software components and their interactions. Such models externalize (and usually visualize) software design in a formal way and can actually be used to test design integrity prior to actually writing code. More recently, object modeling has been applied to create representations of complex systems outside of software engineering, such as the modeling of business processes. [Reference BPML] An object model describes a system in terms of a set of objects and their relationships. Individual objects are described by a set of parameters according to some schema. Relationships amongst objects include (but are not limited to): workflow, logical flow, containership, and inheritance. Figures 1a and 1b demonstrate containership and inheritance, respectively.

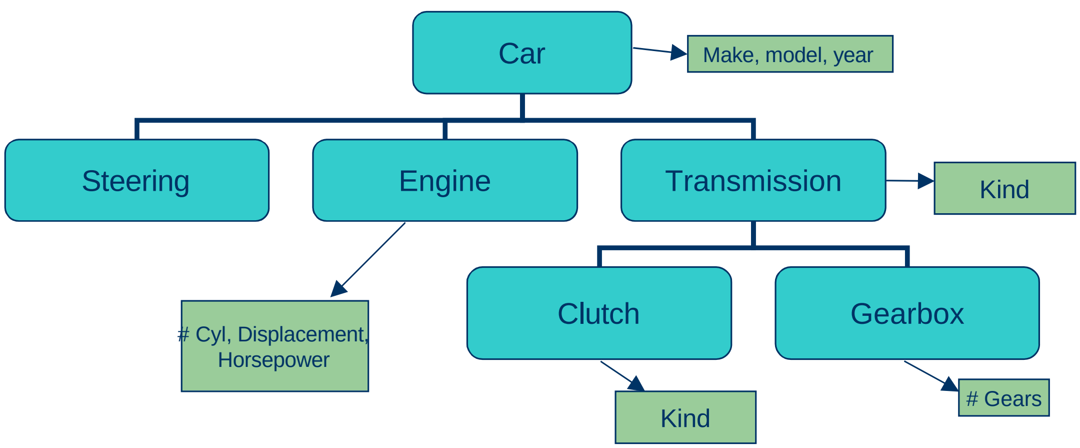

Figure 1a – Component Hierarchy

In Figure 1a, we see a simple model of a car. At the top level we have the gross "car" object, which is described in terms of parameters that describe the car as a whole, such as make, model, and year. The car is further defined in terms of its subcomponents – other objects related by containership. These subcomponents have their own parameters, and potentially relationships to further subcomponents. It is important to note the role of reusability here – the “transmission” object for a car can just as well be used as a component of a truck or a motorcycle. It is a distinct first class object.

The PADI object model uses containership extensively. Conceptually, an assessment argument consists of the KSAs one wants to measure, data in the form of students’ work and procedures for evaluating it, and key features of the task situation which, according to the rationale underlying the task, elicit the targeted KSAs. The PADI Design Pattern object is a schema which defines these attributes more formally, so that a user filling in a design pattern must produce a complete and coherent collection of elements for an assessment argument. The objects in a PADI Template are more complex, with a Template itself a container for other objects such as Activities and Student Models which themselves are containers for objects such as Evaluation Procedures, Measurement Models, Materials and Presentation, and Task Model Variables. While reflecting an assessment argument, these objects are used to specify the more technical elements and properties of implemented assessment tasks. In complex assessments, some of the attributes of these objects (such as Measurement Models and Evaluation Rules) may require specialized expertise to detail. However the structure of the object model allows the details of methods to be encapsulated, so that experts of different phases of assessment can make sure their work coheres at only the level at which objects share relationships, attributes, or information.

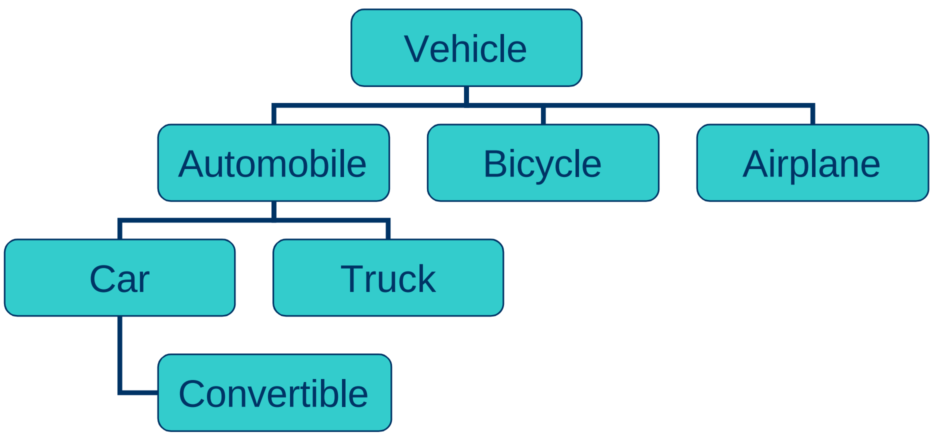

Figure 1b-Inheritance Hierarchy

In Figure 1b, we see an inheritance hierarchy containing a car. The root ancestor of the car is rather abstract – “vehicle”, something that has the capacity to transport other things. As we move down the hierarchy we come to “automobile” which inherits the definition of “vehicle” and adds properties such as – “moves on the ground”, has an engine. “Automobile” is still abstract, when we move one level further we come to the concrete “car” and “truck” objects. Although, we see that even the concrete “car” can be further extended as “convertible”. Inheritance hierarchies convey information about *what* *kind* of object we have. In PADI, the Measurement Model class can be specialized to define more detailed classes of Measurement Models for different psychometric models. All inherit from the Measurement Model class the role of modeling conditional probabilities of observable variables as functions of student model variables. The particular statistical model used and the parameters it requires are defined at the level of subclasses of particular Measurement Models. In addition to containership and inheritance, we might also model flows within a car, such as a functional model of a combustion engine representing the logical and material flow from stroke to stroke of the piston.

Once created, an object model serves many roles in the design process. Foremost, an object model allows users to "concretize" a conceptual design, without fully committing to an implementation. This "proto-implementation" can be tested, whether by the simple visual inspection that "all the parts connect", or more rigorous testing of, say, coherent logical flow. The object model also creates tangible handles on the conceptual design; users have a shared artifact that can be manipulated in specific ways that are clear to all, rather than a more nebulous design document. Lastly, the object model serves the critical role as blueprint to guide implementers, though object models, laden as they are with semantic value, can be far more dynamic than the static drawings that come to mind when we think of blueprints.

As part of the PADI project, object modeling serves two major roles. First, it was a key component of the design of the PADI system itself. More importantly however, the principles of object modeling extend to the use of the PADI system. Conceptualizing components of an argument for assessment design as objects whose properties are inheritable, reusable, etc. brings to bear all the functionality of object modeling to the assessment design process. Moreover, the links among PADI objects contain embedded information about evidence-centered design and navigation among elements of the CAF.

6 Conclusions

We have sought to demonstrate the epistemic nature of the PADI design process. That is, we have attempted to lay out how it is that PADI generates new knowledge for the assessment designer that uses it. After a thorough examination of the theoretical intent of PADI and early practical usage of the system, we have determined several channels of knowledge creation. First and foremost, are the principles of evidence-centered design. Users of the PADI system reap immediate benefits from research in this area, as the principles are "wired in" to the PADI Object Model schema. Because of this, designers cannot help but generate assessment designs with coherent evidentiary reasoning backing them. The linkages among the task model, evidence model, and student model are clearly explicated by the Object Model in such a way as to provide a "map" to completeness for designs in progress.

In conversation with the various design teams, we have discovered that the standardized terminology for assessment components established by PADI is extremely valuable, as design team membership tends to be highly diverse, drawing experts from a variety of domains, each with a different cultural understanding of assessment. The alignment of terms and of conceptual understanding of conceptual components allows teams to work together and reap the benefits of their diverse skills and expertise.

The object-oriented approach to design embraced by PADI provides clear benefits to user's assessment design knowledge. Most importantly, it allows users to reap the greatest benefit from existing work. Rather than starting from scratch, users are able to build on existing work in a number of ways. For one, all existing work in the PADI library is of course designed under the same framework (the PADI Object Model), and the subcomponents of that framework are clearly identifiable, thus making it easy for a user to identify exactly what an existing design does. Furthermore, subcomponents are clearly "cross-referenced" via "part-of" (containership) and "kind of" (ancestry) relationships, thus making it easy to, for instance, find examples of assessment templates that use a particular Student Model. Once an existing item of interest is found, it is then easy to build upon that item in one of two ways. "Abstract" templates can be concretized (extended) by supplying concrete values left vaguely defined or undefined. Or, owing to the clean encapsulation of parts in the PADI system, it is also easy to modify an existing template by swapping out certain subcomponents and swapping in replacements.

The "compare and contrast" game presented in Collins & Ferguson is clearly a factor in the PADI system. Virtually all users reported on the value of being able to easily compare components of their assessments with equivalent components of other assessments and to learn from the contrast. Similarly, many design teams went through a process of "reverse engineering" in the course of their efforts. The reverse engineering process is a decomposition of an assessment into PADI's clear structure. It allows designers to more deeply understand existing assessments, as well as facilitating the creation of variations on those assessments. Lastly, inheritance hierarchies, as well as all of the other Template superstructures discussed in the investigations of emerging practice, allow users to view a given assessment design as it sits in a broader "assessment landscape". This landscape of designs allows designers to understand the extrinsic value of a given design, as well as pointing the way to fertile territory for new assessment design work.

References

PADI Design Patterns Tech Report

Tech Reports for BioKIDS, FOSS, GLOBE, and Floating Pencil (all currently "in preparation." They will be filled in when they are completed)

Kane, M. (1992). An argument-based approach to validation. *Psychological Bulletin,112*, 527-535.

Messick, S. (1989). Validity. In R.L. Linn (Ed.), *Educational measurement* (3rd Ed.) (pp. 13-103). New York: American Council on Education/Macmillan.

Messick, S. (1994). The interplay of evidence and consequences in the validation of performance assessments. *Educational Researcher*, *23*(2), 13-23.

Mislevy, R.J., Steinberg, L.S., & Almond, R.G. (2002). On the roles of task model variables in assessment design. In S.H. Irvine & P.C. Kyllonen (Eds.), *Item generation for test development* (pp. 97-128). Hillsdale, NJ: Erlbaum.

Stewart, J., & Hafner, R. (1994). Research on Problem Solving: Genetics. In D. Gabel (Ed.), *Handbook of research on science teaching and learning* (pp 284-300). New York: Macmillan.

White, B. Y., & Frederiksen, J. R. (2000). Metacognitive facilitation: An approach to making scientific inquiry accessible to all. In J. Minstrell & E. van Zee (Eds.), *Teaching in the inquiry-based science classroom.* Washington, DC: American Association for the Advancement of Science.

Wiley, D.E. (1991). Test validity and invalidity reconsidered. In R. Snow & D.E. Wiley (Eds.), *Improving inquiry in social science* (pp 75-107). Hillsdale, NJ: Erlbaum.

---

[i] Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003) *A Brief Introduction to Evidence-Centered Design*. ETS Research Report RR-03-16.

[ii] Collins, A & Ferguson, W., (1993). *Epistemic Forms and Epistemic Games: Structures and Strategies to Guide Inquiry*. Educational Psychologist, *28* (1), 25-42.

[iii] Bass, K. M., Magone, M. E., & Glaser, R. (2002). *Informing the design of performance assessments using a content-process analysis of two NAEP science tasks* (CSE Technical Report 564). Los Angeles: National Center for Research on Evaluation, Standards, and Student Testing.

[iv] Baxter, G. P., & Glaser, R. (1998). *Investigating the cognitive complexity of science assessments.* Educational Issues and Practice, 17(1), 37-45.

[v] Riconsante, M., & Mislevy, R. (in press). *Introduction to Task Templates (PADI Technical Report #3). Menlo Park, CA: SRI International.*

[vi] http://www.uml.org

[vii] Dahl, O.-J., B. Myhrhaug and K. Nygaard (1970). SIMULA-67 Common Base Language, Norwegian Computer Centre, Oslo, Norway, Pub. **S-22**

[viii] Kay, Allan; (March 1993); *The early history of Smalltalk*; ACM SIGPLAN Notices, Volume 28, Issue 3

Sponsor
The National Science Foundation, Grant REC-0129331

Prime Grantee
SRI International. *Center for Technology in Learning*

Subgrantees
University of Maryland
University of California, Berkeley. *Berkeley Evaluation & Assessment Research (BEAR) Center* and *The Full Option Science System (FOSS)*
University of Michigan. *BioKIDs*