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

John Brecht, SRI International

Robert Mislevy, University of Maryland

Geneva D. Haertel, SRI International

Kathleen C. Haynie, Kathleen Haynie Consulting

Paper presented at the
Annual Meeting of the American Educational Research Association,
April 14, 2005 Montreal, Canada

Abstract

We examine the Principled Assessment Designs in Inquiry (PADI) design system as a complex of epistemic forms with several epistemic games that may be played to complete the forms. 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. This analysis illuminates what new understanding about assessment is generated in the PADI process, thus illustrating the relative value of the PADI process.

Background – PADI

PADI is a design system that guides users through the complex 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 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).[1] The primary tenet of ECD is that assessment design begins with evidentiary reasoning, not the design of tasks. 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.

Background – Epistemic Forms and Games (EF/G)

Collins and Ferguson's Definitions

In their seminal paper[2], Collins and Ferguson (1993) introduce the concept of "epistemic forms and games". Epistemic forms are information structures that guide inquiry. Epistemic games are processes for completing those structures. Examples include structures as simple as lists and as complex as system-dynamics models. (Table 1)

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)

What makes a given form epistemic is the "game" or procedure that is used to complete the form. So, for example, a list that was randomly generated is merely a form – a structured arrangement of the list's components that does not necessarily provide insight into the components' individual or collective nature, whereas adding constraints, such as "members must be distinct and of the class 'mammals'", causes the completed form to represent knowledge – the form can provide immediate answers to specific questions.

Components of Epistemic Games

Components of epistemic games include entry conditions, moves, constraints, and transfers. Entry conditions determine whether it is appropriate and possible to engage in a particular epistemic game, given a particular query and an initial state (available data, assumptions, conceptual design, etc…) Moves are those actions that the player can take such as adding a component to a list. Constraints are limitations on when and how moves may be applied, and as indicated above, are generally the source of the game's epistemic nature. Constraints on listmaking can be simple and local (such as class membership), or they can be higher level global properties (such as distinctness) of items in the list. Lastly, transfers are moves from one game to another, such as the decision to transform a list into a hierarchy. Transfers also include minor modifications to the motivating query – such as changing the question: "How many mammal species are there?" to "How many *non-extinct* mammal species are there?"

The Benefits of EG/F Analysis for PADI

Epistemic games are processes by which unstructured information is transformed into structured information that contains knowledge that can guide 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 an assessment is at its core an exercise in evidentiary reasoning. PADI has sought to bring this fundamental fact to the fore, in terms of explicit and unifying structures that unify previous forms and capture captures the essential elements of all assessment design. 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, nor is it merely 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.

PADI's Epistemic Forms and Games

Inquiries Guided by PADI

The most critical component in the description of an epistemic form and game is the type of inquiry it can guide. In PADI, there are two general kinds of inquiries, corresponding to PADI's two primary forms. For example, creators of Design Patterns could begin to complete the design pattern form with the statement of a standard for a science inquiry skill and seek to answer inquiries of the form:

> *"What are potential constituent elements of assessment tasks that can provide information about this skill, [insert statement of particular skill], as implied by this standard?*

A user of the PADI design system creating an assessment begins with a relatively unstructured conceptual assessment argument, generally in terms of the elements or processes of particular, familiar, kinds of assessments—a notion of the knowledge, skills and abilities they want to measure, for example, the observations they want to collect, or features of the tasks the user think may be relevant, and a query of the following form:

> *"Given this conceptual assessment argument, what are the attributes of a specific assessment task that can elicit, identify, and synthesize evidence in a way that instantiates the assessmentl argument?"*

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

PADI's Principles

First we consider the underlying principles that are "built in" to the PADI system, and that will necessarily underlie all of PADI's forms and games. Second, we will examine specific games played by PADI's users in their design tasks.

Evidence Centered Design

Evidence Centered Design[1] (ECD) is the theoretical framework of PADI. ECD is a set of design principles (and suggested architecture) meant to make evidentiary arguments the basis of assessment design. Evidence centered design is critical in the creation of assessments that rely on technological advances to collect complex data and advances in evidentiary reasoning and statistical modeling to derive evidence from those data for an ever widening set of student knowledge, skills, and abilities of interest to cognitive and instructional science.

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 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, this 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).

As we shall see, the interconnectedness of the components of PADI's object model makes it impossible for designers to design tasks in isolation of measurement considerations. Thus, even a task-centric designer using the PADI system will quickly find her or himself contending with evidentiary implications for the task they are designing.

Object-Oriented Design

Object Modeling is the technological framework of PADI. The components of the Conceptual Assessment Framework[1] have been expressed as an object model schema[3] in the Unified Modeling Language (UML)[4]. 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[5] and Alan Kay's well-known Smalltalk[6]. 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 rather 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.

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. 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.

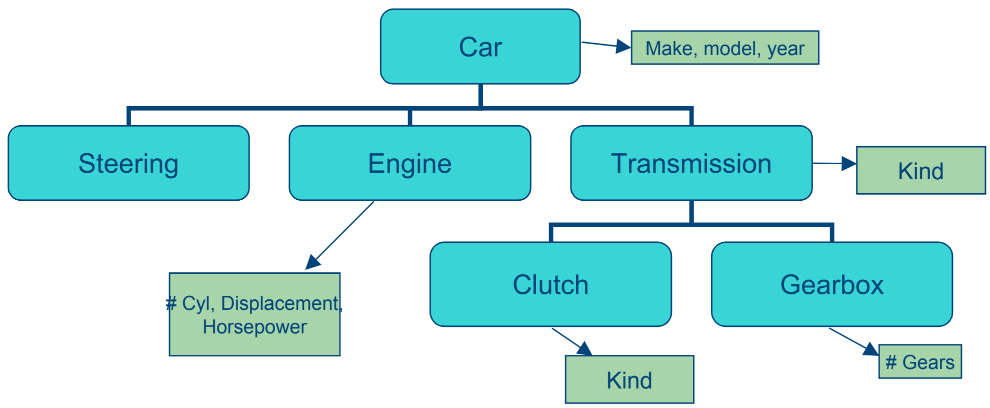

Figure 1a – Component Hierarchy

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.

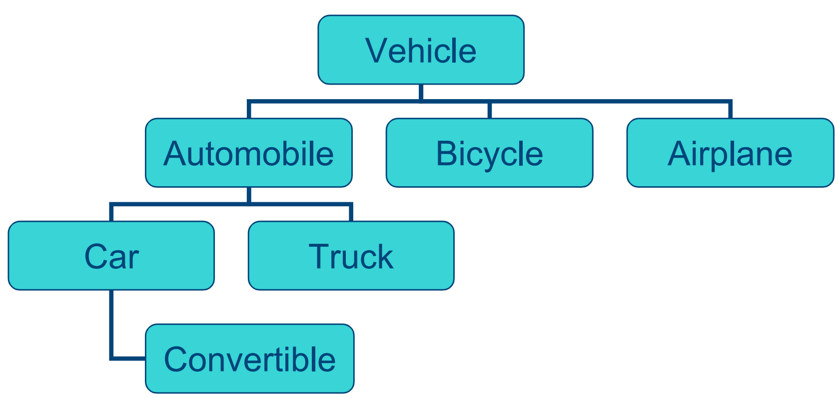

Figure 1b-Inheritance Hierarchy

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”.

Object Modeling as an Epistemic Form and Game

As an epistemic game, object modeling can be seen to be a combination of three games outlined by Collins and Ferguson:

1. The structural analysis game of “compare and contrast by decomposition” – the decomposition is specified by the object schema, and then two or more instantiations of schema object can be compared component by component and parameter by parameter.
2. The structural analysis game of “hierarchy” – as has been shown, many hierarchies of relationships can exist amongst objects.
3. The functional analysis game of “form-and-function” analysis – an object model not only describes the structure of objects but also potentially their structural interrelationships.

The object modeling game occurs in two stages. The first stage is the creation of the schema – the metamodel of the domain. (In object oriented programming, this would be the creation of class definitions.) The inquiry here is something like:

What are the essential kinds of components and relationships of all potential situations in this system or domain?

The target epistemic form in this case is a completed description in UML of potential components of the system and their potential relationships. The game then is to study the domain of interest, determine the components that compose the system and what their essential features are, and then determine the ways those components can interact. The most significant sort of constraints in this game have to do with determining what features are truly “essential” for the modelers’ purposes. A brevity constraint exists, as the model must be wieldy in order to be practical. Simultaneously, a completeness constraint also exists, as the model must be sufficiently complete to represent potential structures and behaviors accurately. Therefore, designers must carefully weigh brevity vs. completeness as each new component is considered. For example, in our car model, perhaps we do not need to specify every nut and bolt in the vehicle, as all of those trivial specifications might “crowd out” more important features. Perhaps our car model does not even need to contain the cars components. A list of cars for sale in a classifieds section of a newspaper may only require a model that includes make, model, year, color, and perhaps an overall “condition” parameter.

The completed epistemic form of the first stage becomes the target epistemic form for the second stage. Once a schema has been created, modelers may then instantiate objects of interest in the schema’s terms. Real or conceptual objects or systems to be modeled must be examined and valuated relative to the schema. Values are then assigned to the schema’s parameters, and relationships between instantiated objects are defined. In this stage, virtually any inquiry about the domain may be pursued. In turn, a wide variety of game constraints are possible depending on the inquiry being pursued. Some constraints will be implicitly embedded in the schema design itself, for example the degree of specificity of the models.

The PADI Object Model

Now let us proceed to consider PADI as a special case of the Object Modeling game. The PADI design system was intended to guide inquiries regarding appropriate evidence-driven assessment designs for particular inquiry skills. Assessment designs are complex 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 the object modeling game – the creation of a schema for the domain. PADI's schema was designed based on Mislevy et al's previous work on the Conceptual Assessment Framework[1]. The schema is presented visually in Figure 2.

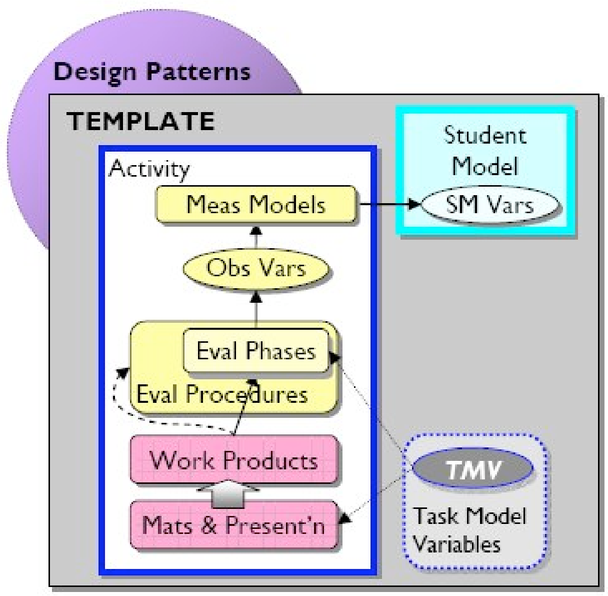

Figure 2 – The PADI Object Model Schema

As mentioned before, the two primary products of the PADI Design System are Design Patterns and Templates. Figure 2 shows Design Patterns 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. Some of the parameters of Design Patterns include: Focal Knowledge, Skills and Abilities; Rationale; Potential observations; and Potential work products.

Templates on the other hand have significant internal structure, as well as potentially being part of an external structure of other Templates primarily through inheritance relationships, but sometimes as actual subcomponents. A given Template links to one Student Model, which in turn links to one or more student model variables. The Student Model is an attempt to model internal Knowledge, Skills, or Abilities with respect to a particular domain. The Activity links to one Task Model, which links to its own set of Task Model Variables. Task Models represent ways in which the assessment task can be varied, for example "difficulty level".

A Template may contain one or more Activities, which are partially parameterized by Task Model Variable settings and that link evidence produced by the Measurement Models to Student Model variables. Internally, Activities model task workflow and logical flow, beginning with Materials and Presentation given to students. Students then produce particular Work Products. Work Products are fed into Evaluation Procedures, which may have one or more internal Evaluation Phases. Evaluation Phases produce Observable Variables. Observable Variables are passed to Measurement Models, which use statistical techniques to relate Observable Variables to Student Model Variables.

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. Children 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.

The PADI Design Process

The second phase of Object Modeling is the actual creation of working models based on the schema generated in the first phase. This is the epistemic game played by users of the PADI system as they create their assessments or families of assessments. Entry conditions in the PADI design game relate directly to the inquiry being pursued. If the user is seeking to express the substance of an assessment argument or a general practice that could apply to many assessments of some genre, then they are 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 have effectively been made ahead of time in schema design. The user's task then is to populate those elements in their instantiation. The creation of a Design Pattern is more an aesthetic task than a mechanical one. The designer's goal is to capture the substance of an assessment argument such that it may be used 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 is in fact an abstract 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, and the Measurement Model cannot be complete without ultimately linking it to a Student Model. Thus, the PADI Template Schema insures that all designers must consider their evidentiary assessment argument before long.

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. 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. Here we give an overview of the results of this study. In a future technical report, we will examine the results more thoroughly.

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 currricula 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 the value of Object-Oriented design principles, particularly reuse. 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 designer 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 intitial Design Pattern, using it to record high-level ideas that emerged 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, see other potential observable variables.

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.

The GLOBE team began their work with a template of sorts already, that had been designed under previous GLOBE work. This previous work on the GLOBE Template already 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 also were able to pick a set of existing GLOBE activities to test those skills. 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 and vary grade level with Task Model Variables. Still, there were other dimensions along which the team wanted to vary their activities, 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:

- Multiple inquiry, multiple content
- Multiple inquiry, single content
- Combined inquiry and content (in one Student Model Variable)
- 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."*

BioKIDS

BioKIDS is another of the curriculum partners selected for participation in the PADI Project. (See http://groundhog.sprl.umich.edu/site/biokids.html). 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 three curricula units with different subject matter, which they distilled down to a core set of inquiry skills. 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 at the very beginning, 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 matrix that varied along content and inquiry. This conceptual structure was based on previous design work involving a similar matrix of tasks. The BioKIDS matrix is presented in Table 1.

The team worked on their 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.

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.

Floating Pencil

The Floating Pencil team sought to reverse engineer a 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 were 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. 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 psychometrically parallel versions of the task. Different bases for the task family was explored including scaffolding levels, a set of already existing constructs (e.g., science inquiry standards, phases of inquiry), and task characteristics of already existing performance assessment. PADI forms were considered in conjunction with this: Should the family of tasks be defined as a Template or Design Pattern? What are the advantages of each? Could an abstract template capture both the detail and breadth needed for a Floating Pencil task family?

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. Second, 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 "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.

---

[1] Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003) *A Brief Introduction to Evidence-Centered Design*. ETS Research Report RR-03-16.
[2] Collins, A & Ferguson, W., (1993). *Epistemic Forms and Epistemic Games: Structures and Strategies to Guide Inquiry*. Educational Psychologist, *28* (1), 25-42.
[3] Riconsante, M., & Mislevy, R. (in press). *Introduction to Task Templates (PADI Technical Report #3). Menlo Park, CA: SRI International.*
[4] http://www.uml.org
[5] Dahl, O.-J., B. Myhrhaug and K. Nygaard (1970). SIMULA-67 Common Base Language, Norwegian Computer Centre, Oslo, Norway, Pub. **S-22**
[6] Kay, Allan; (March 1993); *The early history of Smalltalk*; ACM SIGPLAN Notices, Volume 28, Issue 3