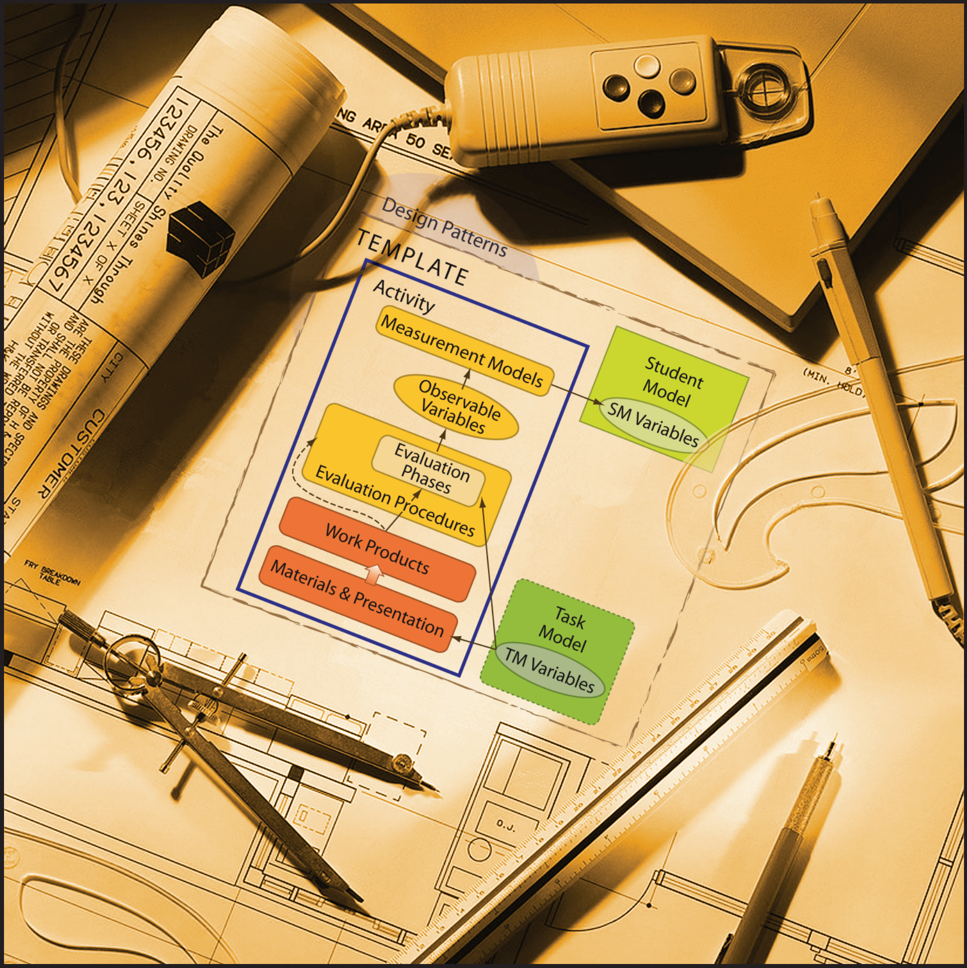


PADI Technical Report 19 | January 2008

Design Rationale for an Assessment Task Authoring System: A Wizard for Creating "Mystery Inquiry" Assessment Tasks

PADI | Principled Assessment Designs for Inquiry

Lawrence Hamel, CodeGuild, Inc.
Robert J. Mislevy, University of Maryland
Fielding I. Winters, University of Maryland

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

PRINCIPLED ASSESSMENT DESIGNS FOR INQUIRY
TECHNICAL REPORT 19

Design Rationale for an Assessment Task Authoring System: A Wizard for Creating "Mystery Inquiry" Assessment Tasks

Prepared by:
Lawrence Hamel, CodeGuild, Inc.
Robert Mislevy, University of Maryland
Fielding I. Winters, University of Maryland

Acknowledgments
This material is based on work supported by the National Science Foundation under grant REC-0129331 (PADI Implementation Grant). We are grateful for contributions by Geneva Haertel, Cathleen Kennedy, and Futoshi Yumoto.

Disclaimer
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

CONTENTS

FIGURES

TABLES

ABSTRACT

Based on the Principled Assessment Designs for Inquiry (PADI) project, as supported by the National Science Foundation to improve the assessment of science inquiry, this report describes an approach to supporting the authoring of assessment tasks. In keeping with the PADI emphasis on complex tasks, we focus on an example that concerns families of iterative, problem-solving tasks exemplified by the well-known Mystery Powders investigations. An implementation of Mystery Powders was previously designed and implemented through the PADI framework, and taking that as a target, we show how an online interview—i.e., a wizard—might provide support for authoring a principled assessment task while hiding much of the complexity behind that assessment design.

1.0 Introduction and Overview

Principled Assessment Designs for Inquiry (PADI) is a project supported by the National Science Foundation to improve the assessment of science inquiry. The PADI project has developed a design framework for assessment tasks, based on the evidence-centered design (ECD) framework introduced by Mislevy, Steinberg, and Almond (2003). PADI was developed as a system for designing blueprints for assessment tasks, with a particular eye toward science inquiry tasks—tasks that stress scientific concepts, problem solving, building models, using models, and cycles of investigation. The PADI framework guides an assessment developer's work through design structures that embody assessment arguments and take advantage of the commonalities across the assessments for sharing and re-using conceptual and operational elements (Mislevy & Haertel, 2006).

ECD seeks to integrate the processes of assessment design, authoring, delivery, scoring, and reporting. Work in PADI, however, focused on design layers that lie above the level of specific environments for task authoring and assessment delivery. This report describes an approach to developing task-authoring support that builds on the higher level PADI *task templates.*

To illustrate this approach, we provide an example of the process involved with authoring an assessment. In keeping with the PADI emphasis on complex tasks, the example concerns families of iterative problem-solving tasks exemplified by the well-known "Mystery Powders" investigations (e.g., Baxter, Elder, & Glaser, 1995). We build from the computer-based implementation of Mystery Powders tasks designed through the PADI framework as described in Siebert et al. (2006).

The next section (Section 2.0) provides background on ECD and PADI, with an emphasis on the layers of design, implementation, and delivery in assessments and the PADI structures that support this work. Task authoring systems are introduced and illustrated with three existing examples. Section 3 provides the rationale for the present project. Section 4 describes the project as implemented, including the use cases that motivated and helped structure the work. Section 5 walks through examples with a series of screen shots utilizing two of the use cases to contextualize the example. Section 6 discusses the benefits we see of this line of research.

2.0 Background

2.1 Evidence-Centered Design

The PADI project uses an ECD approach to building blueprints for assessments. Central ideas in ECD are the assessment argument, layers of the assessment, and the role of knowledge representations in designing and implementing assessments. Messick (1994, p. 16) concisely lays out the key aspects of an assessment argument by asking, "what complex of knowledge, skills, or other attributes should be assessed. Next, what behaviors or performances should reveal those constructs, and what tasks or situations should elicit those behaviors?" All of the many terms, concepts, representations, and structures used in ECD aim to support constructing a coherent assessment argument and building machinery to implement it.

Using a "layers" metaphor from architecture and software engineering, ECD organizes the design process in terms of the following layers: domain analysis, domain modeling, conceptual assessment framework, assessment implementation, and assessment delivery (Mislevy & Riconscente, 2006). The fundamental work in assessment design can be viewed as creating, transforming, and using information within and between these layers. Table 1 summarizes these layers in terms of their roles, key entities (e.g., concepts and building-blocks), and the external knowledge representations that assist in achieving each layer's purpose; cycles of iteration and refinement across layers are the norm. (For more information on the role of knowledge representations in ECD, see Mislevy et al., 2007)

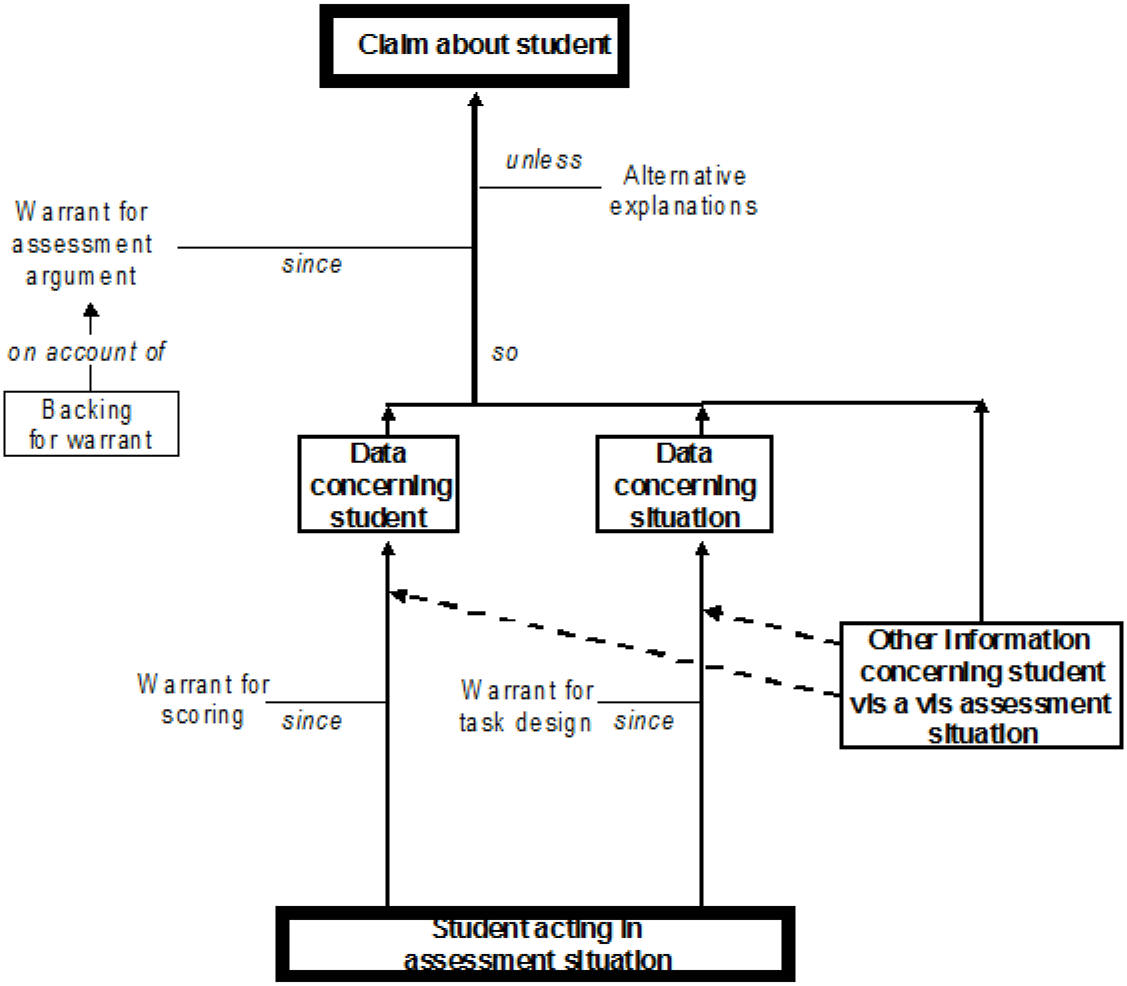
The first layer in the process of designing an assessment is the *domain analysis*. It lays the foundation for later layers by defining the knowledge, skills, and abilities (KSAs) that assessment users want to make inferences about, the student behaviors they can base their inferences on, and the situations that will elicit those behaviors.

The next layer in the ECD process is *domain modeling*. *Domain modeling* structures the outcomes of *domain analysis* in a form that reflects the narrative structure of an assessment argument (Figure 1), in order to ground the more technical models in the next layer. We will see that PADI uses objects called *design patterns* to assist task designers with *domain modeling*.

Table 1. Layers of Evidence-Centered Design

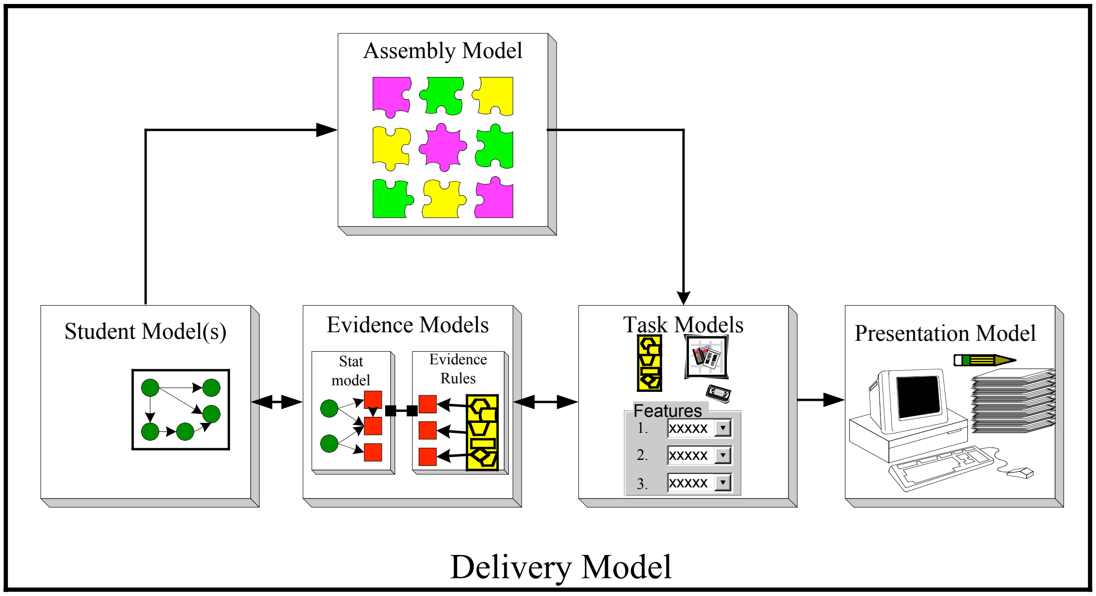
Layer	Role	Key Entities	Selected External Knowledge Representations
Domain Analysis	Gather substantive information about the domain of interest that has direct implications for assessment: how knowledge is constructed, acquired, used, and communicated.	Domain concepts, terminology, tools, knowledge representations, analyses, situations of use, patterns of interaction.	Content standards, concept maps (e.g., Atlas of Science Literacy, AAAS, 2001). Representational forms and symbol systems of domain of interest, e.g., maps, algebraic notation, computer interfaces.
Domain Modeling	Express assessment argument in narrative form based on information from Domain Analysis.	Knowledge, skills and abilities; characteristic and variable task features, potential work products and observations.	**Design patterns,** assessment argument diagrams, content-by-process matrices
Conceptual Assessment Framework	Express assessment argument in structures and specifications for tasks and tests, evaluation procedures, measurement models.	Student, evidence, and task models; student model, observable, and task model variables; rubrics; measurement models; test assembly specifications.	**PADI task templates;** test specifications; algebraic & graphical EKRs of measurement models; item generation models; generic rubrics; automated scoring code.
Assessment Implementation	Implement assessment, including presentation-ready tasks, scoring guides or automated evaluation procedures, and calibrated measurement models.	Task materials (including all materials, tools, affordances); pilot test data for honing evaluation procedures and fitting measurement models.	**Task objects**; coded algorithms to render tasks, interact with examinees, evaluate work products; tasks as displayed; IMS/QTI representation of materials; ASCII files of parameters.
Assessment Delivery	Coordinate interactions of students and tasks: task-and test-level scoring; reporting	Tasks as presented; work products as created; scores as evaluated.	Renderings of materials; numerical and graphical score summaries; IMS/QTI results files

Figure 1. An Assessment Argument Diagram (after Mislevy, 2006)



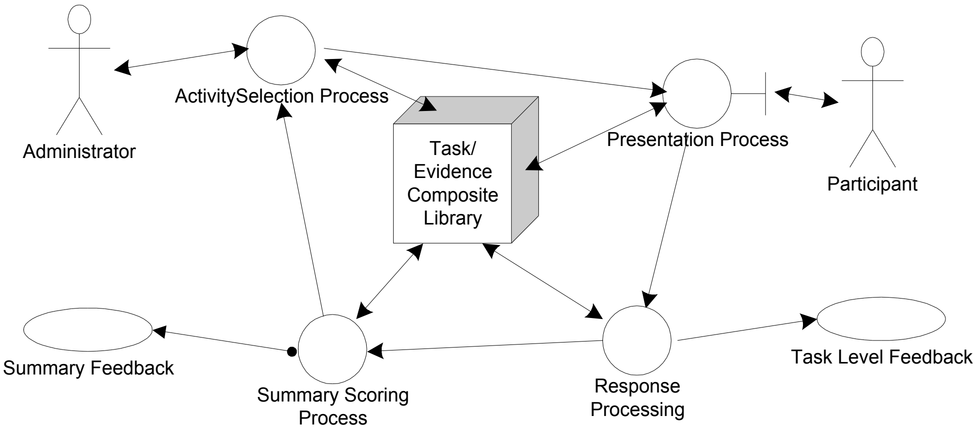
Next, the *conceptual assessment framework (CAF)* articulates the technical specifications for the materials and processes that embody assessments. In particular, the assessment argument that was laid out in narrative form at the *domain modeling* layer is expressed in the CAF in terms of specifications for tasks, measurement models, scoring methods, and delivery requirements within *templates*. The central models within the CAF are the Student Model, Evidence Model, and Task Model (Figure 2). In addition, the Assembly Model determines how tasks are assembled into tests, the Presentation Model indicates the requirements for interaction with a student (e.g., simulator requirements), and the Delivery Model specifies requirements for the operational setting. Details about task features, measurement-model parameters, stimulus material specifications, and the like are expressed in the CAF model *templates* in terms of knowledge representations and data structures, which guide their implementation and ensure their coordination. These *templates* are thus essentially blueprints that specify, at a meta-level, the necessary element for tasks (Mislevy et al., 2007). The PADI system uses *templates*, described in the next section, as its specific form for expressing CAF models.

Figure 2. The Central Models of the Conceptual Assessment Framework (CAF)



Delivery of an assessment from an ECD perspective is defined by a four-process architecture that includes the *activity selection process*, the *presentation process*, *response processing*, and the *summary scoring process* (Almond, Steinberg, & Mislevy, 2002; see Figure 3). This schema provides a common architecture for delivery of assessments that can be tailored depending on the purpose of the assessment. The four processes described below can be carried out via various means, such by humans, a computer system, or a combination of both. Importantly, the information from the models laid out in the CAF describes the nature and operations of the processes in a particular assessment and the contents of the messages that the processes use, create, and transmit for the assessment to serve its purpose.

Figure 3. The Four-Process Delivery Architecture



The first process, *activity selection*, entails selecting and sequencing the tasks and deciding where to start and when to stop. To this end, the *activity selection* process monitors the state of the assessment by receiving communication from the other processes to monitor the participants' state; it carries out the strategy of the assessment, articulated in the Student Model Variables in the CAF and constrained by the Task Model Variables in the CAF, by choosing tasks that will maximize information for the particular strategy; it customizes the strategy for particular circumstances, such as participants with special needs; and it responds adaptively to the various other components in the architecture by monitoring these components.

The *presentation process* presents the tasks to the participant and sends the participant's responses in the form of Work Products. The *presentation process* is largely dictated by the Task Model and Presentation Material for each task selected by the *activity selection* process. The *presentation process* must locate and present different stimulus media, such as images or sound files, from a multimedia database or server; it must capture the participant response and put it into Work Products, as specified by the Task Model; it must manage the tools used for building the interface, including scrolling and buttons, calculators and dictionaries, and simulators and word processors; it is responsible for the layout of the information the participant sees as part of the task; and it must respond to messages from the *activity selection* process.

The *response processing* receives participant Work Products (responses) from the *presentation process* and begins the scoring cycle. The *response processing* is responsible for implementing the instructions contained in the Evidence Rules—specified in the Evidence Model in the CAF—that dictate how to score responses for particular tasks. Different Evidence Models can exist for a given task, when the purpose of the assessment varies. In particular, the *response processing* employs the Evidence Rules to score Work Products, and in so doing, it sets the values of the observables and sends them to the *summary scoring process*.

The *summary scoring process* updates the Scoring Record once the values of the observable variables have been sent to, it. The Scoring Record represents the current belief about the participant's knowledge, skills, and abilities in the form of a probability distribution. However, other common psychometric models such as percent correct, weighted number correct, Bayesian formulation, and graphical models also fit into this framework.

2.2 *PADI*

A central goal of the PADI project is to create a process that makes assessment design and delivery streamlined and efficient, by providing flexibility, interoperability, and reusability in its structures. As such, the conceptual framework developed in PADI defines a system of interrelated objects which contain assessment information, including *design patterns*, *templates,* and *task specifications*.

2.2.1 Design Objects in the PADI Design System (PDS)

Design patterns lie at the Domain Modeling layer of the ECD framework. They provide a narrative way to express the assessment argument and the relationships between different models in the framework. Examples of PADI *design patterns* address model-based reasoning, design under constraints, troubleshooting finite systems, and building scientific explanations. For more details on the role and structure of *design patterns* in PADI, see Mislevy et al. (2003).

Templates are forms used in the CAF layer of ECD. They embody Student, Evidence, and Task Models, and as such, contain more specific and technical information than *design patterns*. They combine task environment information with evidence evaluation logic for creating tasks. As Luecht (2002) describes, *templates* use data structures to instantiate the goals of the assessment, and their power lies in the ability to adapt and reuse the *templates* for various assessment purposes. The attributes, or classes of information, that comprise *templates* provide the information needed to implement assessment tasks, like blueprints for a building. Continuing the analogy, the *templates* contain information for building the elements that are needed to deliver and score actual tasks in the environment of a specific assessment system. This paper describes technology supports for implementing tasks in accordance with PADI *templates*. For more details on the role and structure of *templates* in PADI, see Riconscente, Mislevy, and Hamel, 2005.

Templates contain 23 separate attributes, many of which are relations to other objects, each with their own sets of attributes. Table 2 presents a list and definitions of these *template* attributes. Attributes of the *template* generally retain some flexibility, such as Task Model Variables and Materials and Specifications, that have not been specified yet. In this way, multiple tasks can be authored from the same *template*. When every variable in a *template* is decided and specified for a particular assessment, the *template* becomes a *task specification*, which contains the information for implementing one specific assessment task. To help developers manage the complexity underlying the formulation of a *template* with its attendant attributes, the PADI project developed a Web-based editor for creating and editing *templates*.

Table 2. Template Attributes

1)	"**Type**" attributes indicate whether the object is a finished, complete, concrete Task Specification or a Template which is abstract and general.
2)	"**Student Model Summary**" attributes describe in narrative form the student models in the template.
3)	"**Student Models**" attributes are associations with (potentially shared) objects that contain variables used for accumulating information about a student, and probability distributions associated with these variables.
4)	"**Measurement Model Summary**" attributes describe the nature and requirements for measurement models used in this template. For example, one could mention whether the template requires a multidimensional model, or whether items have dependencies.
5)	"**Evaluation Procedures Summary**" attributes describe a general outline of requirements for evaluation procedures.
6)	"**Work Product Summary**" attributes describe an outline of the things created by the student.
7)	"**Task Model Variable Summary**" attributes describe an outline of all the task model variables that are used by this template.
8)	"**Template-level Task Model Variables**" attributes are associations with (potentially shared) objects that describe features of tasks, as they may be required for task construction, presentation, evaluation of performance, specification of measurement model parameters, and so on (Mislevy, Steinberg, & Almond, 2002)
9)	"**Task Model Variable Settings**" attributes are the exact choices made from among those allowed for each task model variable (TMV). In other words, the designer has specified a given task model variable, and it is no longer variable once its value has been set: The template is "pinned" to use this setting. Settings apply to the template/TMV combination. The same TMV may have different

	settings in different templates if it is associated with more than one template. Templates may also have associated Activities, and these Activities may have associated TMVs, but any setting for an "activity" TMV is still controlled by the template. Settings apply to the template, not to individual Activities, even though a TMV may show up under the Activity only.
10)	"**Materials and Presentation Requirements**" attributes specify how the stimuli are presented to the student and any large-scale needs like having a large room.
11)	"**Template-level Materials and Presentation**" attributes are associations with (potentially shared) objects concerning materials and interaction patterns as arise in the presentation of the task to the examinee.
12)	"**Materials and Presentation Settings**" attributes are the exact choices made from among those allowed for each Materials and Presentation (M&P) item. In other words, the designer has specified a given Materials and Presentation choice, and it is no longer variable. The template is "pinned" to use this setting. Settings apply to the template/M&P combination. The same M&P may have different settings in different templates if it is associated with more than one template. Templates may also have associated Activities, and these Activities may have associated M&Ps, but any setting for an "activity" M&P is still controlled by the template. Settings apply to the template, not to individual Activities, even though an M&P may show up under the Activity only.
13)	"**Activities Summary**" attributes are an overview of all the activities included.
14)	"**Activities**" attributes are associations with (potentially shared) “activity” PADI objects—themselves composite objects that contain information about task properties , stimulus materials, evaluation rules, measurement models, and so on, for a phase of activity in a task. Simple tasks have just one “activity,” but a multi-stage investigation could have several activities.
15)	"**Tools for Examinee**" attributes are things provided to or permitted for use by the examinee.
16)	"**Exemplars**" attributes are associations with (potentially shared) objects that provide examples of the kinds of task that are motivated by this template.
17)	"**Educational Standards**" attributes are associations with (potentially shared) objects of type educational standards, such as those developed by the National Research Council (2002) or state departments of education.
18)	"**Design Patterns**" attributes are associations a template has with (potentially shared) objects of this type.
19)	"**I am a kind of**" attributes are associations with other objects that are more abstract or more general than this object. For example, a dog is a specific kind of animal.
20)	"**These are kinds of me**" attributes are associations with other objects that are more concrete or more specialized than this object. For example, animal is a general category that includes specific kinds of dogs.
21)	"**These are parts of me**" attributes are associations with other objects that contain or subsume this one. For example, a windshield is a part of an automobile.
22)	"**Online resources**" attributes are relevant items that can be found online (URLs).
23)	"**References**" attributes are notes about relevant items, such as academic articles.

It will be noted that many of the objects in the PADI *template* are themselves composite objects; that is, they have internal structures, containing other attributes and objects that are used in specifying an assessment. Table 3 presents the attributes of an Activity.

Table 3. Activity Attributes

1)	**“Measurement Models”** attributes are associations with (potentially shared) Measurement Model objects
2)	**“Evaluation Procedures”** attributes are associations with (potentially shared) Evaluation Procedures (rubric) objects
3)	**“Work Products”** attributes are associations with (potentially shared) Work Product objects
4)	**“Materials and Presentation”** attributes are associations with (potentially shared) Materials and Presentation objects
5)	**“Presentation Logic”** attributes specify the order in which various materials should be presented and algorithmic logic that describes any desired looping or conditional presentation.
6)	**“Task Model Variables”** attributes are associations with (potentially shared) Task Model Variable objects
7)	**“Design Patterns”** attributes are associations with (potentially shared) Design Patterns.
8)	**“Online Resources”** attributes are relevant items that can be found online (URLs).
9)	**“References”** attributes are notes about relevant items, such as academic articles.

2.2.2 The Focus of the PADI Design System (PDS)

The focus of the PADI project was to develop a conceptual framework supporting assessment activities for assessment designers working at the highest level. This included building technology tools such as *templates* and wizards for creating the conceptual grounding for a subsequent assessment. The scope of the project remained at the conceptual level rather than focusing on specific assessment authoring and delivery. Thus, the PADI project was not intended for creation of an actual assessment design system, much less task authoring, although it can structure the conceptual design work to support implementation and delivery activities such as task authoring and automated scoring.

Furthermore, the interface developed under the PADI project was intended for use by assessment experts rather than for direct use by teachers and practitioners. However, under the PADI and ECD approach, a well-developed conceptual framework is necessary for creating a design system and for authoring actual assessment tasks. The idea is that the underlying PADI object model should be general enough to support the design of different kinds of assessments. More specialized interfaces can be developed to focus on particular kinds of assessments and particular kinds of users so they can work in more “user friendly” ways with the same underlying data structures. As such, the work by PADI lays a solid foundation for the authoring system design discussed in this paper.

2.2.3 The Direct Use of PADI

The PADI project has been presented in numerous forums. When we discuss PADI and its conceptual framework, the audience often wants to know how to apply PADI to many different arenas, including direct use by teachers and practitioners. As discussed, the work in PADI to this point has been focused primarily on the underlying conceptual framework and its attendant structures. In this report, we hope to make PADI more accessible to all by showing how an assessment design could flow into an authoring system and make the authoring of a principled assessment relatively easy within a given context. The goal of this report is to provide an example of this process using an assessment, Mystery Powders, that already has been fleshed out in a more conceptual way in the PADI system.

2.3 Examples of Existing, Freely Available, Assessment Authoring Systems

Given the requests by practitioners for very direct assistance in authoring assessments, we review several existing tools readily available to practitioners. The following three systems are available at no cost, in both source code and ready–to–install packages. All are learning systems that typically would be set up by an institution and shared by all their instructors. These tools make life easier for practitioners by helping construct multiple–choice, short–answer, and true–false quizzes easily. The systems also can deliver tests to students via computer screens. Each student's attempted answer can be recorded and usually automatically scored. The teacher can choose whether to give feedback per response, or to show correct answers. These systems generally support the creation of a pool of items that can be aggregated into different groups of items for different purposes.

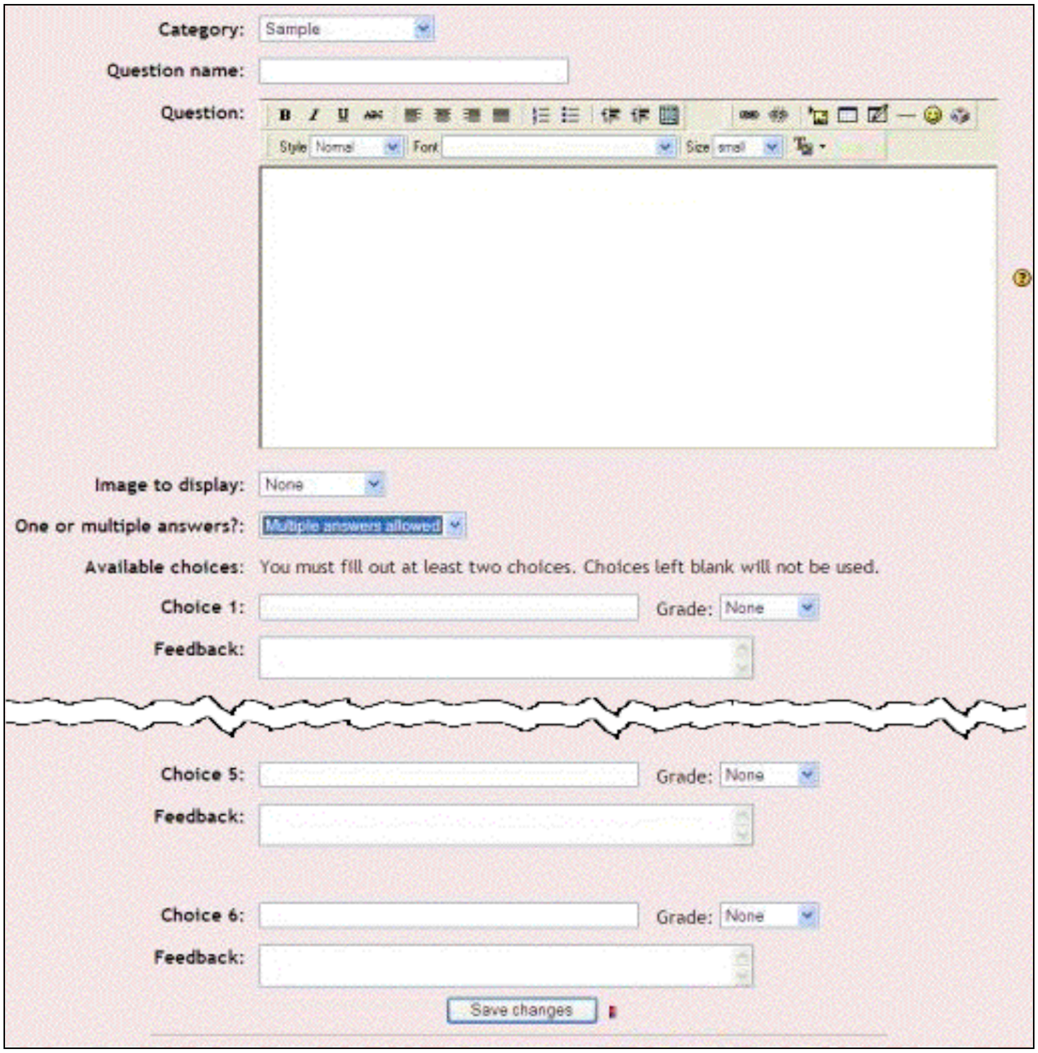
After creating test items in one of these systems, most of the systems permit some kind of export of the items into a format that is largely compatible with other systems, called Question and Test Interoperability (QTI) 1.2 format. Its standards are specified by the IMS Global Learning Consortium, Inc. (http://www.imsglobal.org/question/), and they allow for interoperability between different test components, such as authoring tools, item banks, and assessment delivery systems. The systems' abilities vary with regard to importing items from external sources. Below we briefly highlight three freely available systems.

Moodle

The Moodle system (http://moodle.org/) includes a facility for creating items in a pool, permitting the tagging each item with keywords or categories that assist with aggregation.

The Moodle Quiz module allows an instructor to author items consisting of multiple–choice, true-false, and short-answer questions. For example, creating a new item is shown in Figure 4.

Figure 4. Add Item in Moodle



Category: Sample

Question name:

Question:

Image to display: None

One or multiple answers?: Multiple answers allowed

Available choices: You must fill out at least two choices. Choices left blank will not be used.

Choice 1: Grade: None

Feedback:

Choice 5: Grade: None

Feedback:

Choice 6: Grade: None

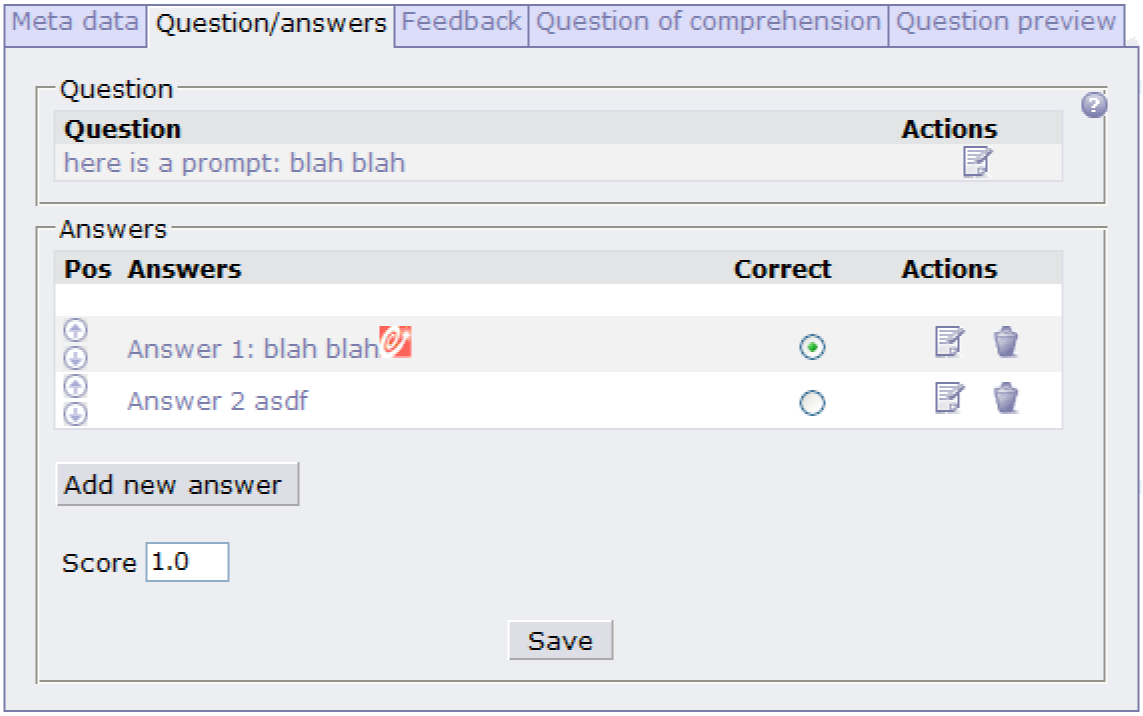
Feedback:

This shows input of a text prompt with standard choices of text styling, along with an image prompt if desired. Possible student answers are specified as text. For each answer, feedback can be delivered. Each response can have a score associated with it (the "Grade" menu). While Moodle cannot import from QTI at the time of this writing, it can export to QTI 1.2 format.

OLAT

The OLAT system (http://www.olat.org/) also includes a system for entering questions for a quiz that can be delivered online and subsequently scored. Figure 5 shows the screen for entering answers to a multiple–choice item. OLAT offers similar facilities to Moodle, with the addition of images and sounds for questions and answers. OLAT can import and export to QTI 1.2 format.

Figure 5. Add Item for OLAT



Question

Question	Actions
here is a prompt: blah blah	

Answers

Pos	Answers	Correct	Actions
	Answer 1: blah blah	●	
	Answer 2 asdf	○	

Score 1.0

Sakai

The Sakai system (http://sakaiproject.org/) includes features like Moodle for creating items with simple prompts and text answers as part of a pool of items. It can deliver tests, with optional feedback, and subsequent scoring of items as well. Sakai can import and export to QTI 1.2 format.

While the three systems discussed here facilitate the construction of assessments as currently configured, they do not support the creation of an assessment argument to support the validity of an assessment, as outlined by the PADI project. For example, none of the systems offer a means to enter a Student Model or connect items with such a model. Psychometric measurement in these systems appears to be focused solely on single-dimensional Student Models.

Further, the systems do not support items beyond the standard types. For example, the systems cannot handle an extended response when students explore and interact with a simulation, where each subsequent interaction would be dependent on previous choices of the student. The systems depend on the independence of all questions from one another.

While these systems do not contain any of the conceptual underpinnings espoused in PADI, they can be used within a PADI framework if they are utilized in concert with the objects and concepts in PADI. As such, we find it useful to describe several of these systems, to both highlight the differences in approach between existing software systems and PADI, as well as to demonstrate ways that PADI can dovetail with existing software to facilitate direct application of PADI principles.

3.0 Project Goals and Constraints

3.1 Initial Goals

Having drafted a framework for the design of *templates* as well as an example of a four-process delivery system, we turned to thinking about an authoring system. Like the delivery–system example, this was beyond what was promised in the grant, and the remaining budget was limited. However, we anticipated value in even a rough design of an authoring system based on PADI principles.

First, we brainstormed what we wanted to see in an authoring system and from this listed our initial goals for an authoring system. Ideally, we wanted the authoring system to

- Accept *task specifications* from PADI Design System (PDS);
- Guide development according to a *task specification* (heed constraints);
- Assist repetitive creation/editing of items;
- Provide a means to create modified Question-test Interoperability XML (QTI+), for development of presentation (e.g., allow the specification of prompts in text and/or image format and allow the creation of multiple–choice responses and other response types);
- Provide a means to create rubrics and related metrics, including descriptions for response categories, for evaluation;
- Provide output primarily in QTI+, Measurement Models, and rubrics; some of these outputs fit within the *template* document defined by PDS; and
- Visualize "coverage" of constraints, such as how all items are distributed on specific design dimensions specified like SMV loading, simple/complex, response types, etc.

3.2 Specific Goals

Following this initial list, we elaborated on what we wanted to show in the first draft of the design of an authoring system and narrowed down our initial goals for actual implementation. These specific goals are instantiated in this paper through mock–ups and the design rationale we provide. The following are specific goals we think are most critical to its development and presentation of a design system. The design of the authoring system must:

- Use PADI design information. The authoring system must utilize the conceptual principles laid out in *design patterns*, *templates*, and *task specifications*, to ensure the continuity and flow between the layers of the assessment process.

- Enforce constraints that flow down from design. For example, if the design specifies that there will be between two and four mystery elements, the authoring system should require the minimum elements be created while preventing more than the maximum from being created.

- Hide or fill in pre–specified items. For example, if the design specifies that a certain format is required, like a multiphase interaction before a final answer, we should hide from the author any other format choice.

- Contain links back to the design system for convenience and deeper understanding. Within the authoring wizard, hyperlinks provide a connection between the specification in the design system and its realization within the authoring system. When constraints and recommendations seen in the authoring system offer links into the design, authors will have a convenient means to explore the assessment more deeply and to understand the design and its assessment rationale while creating the assessment.

- Demonstrate authoring of a complex task. The existing systems reviewed above provide a means to define multiple-choice questions with relatively little variation. The result is tasks where the prompt is displayed, and the student is expected to enter an answer. To differentiate from these authoring systems and demonstrate how a complex interaction might be supported, we can choose a complex task that doesn't fit well into a traditional prompt–and–select type of assessment.

- Hide complexity. A fundamental goal is to enable authoring by practitioners who are not expert in assessment. We can do this with the constraints and hiding of complexity mentioned earlier, as well as other measures.

- Scaffold the process with a wizard. A wizard is an interview process where the author is asked to focus on only one question at a time. Section 5 describes a wizard for the Mystery Powders assessment in greater depth. (For more information on wizards, see Hamel and Schank, 2005.)

3.3 Outside the Scope Goal: Select and Customize from Pools of Pre-Existing Assessments

One ambitious goal would be to complete a large number of sample assessments that authors could browse, clone, and modify, offering a wide array of formats and complexity. However, such extensive development is outside the scope of this small section of the current project. In order to accomplish the most possible with our limited resources, we need to establish criteria for selecting a family of tasks for which we will provide authoring. This family of tasks need to

- Already have a complete PADI design;
- Be generalizable to large set of tasks;
- Allow representation, such as through a wizard, that would hide significant complexity (and thus provide more "bang for the buck"); and
- Have potentially high popularity and usage

4.0 Project Approach

4.1 Use-cases

We drafted the following use-cases from the experience of team members who had worked with the PADI framework and our presumptions about how an authoring system should be created and used. The first use case about Graham, provides background on the motivation for writing this report. The subsequent use cases provide examples of how an authoring system built within the PADI framework can adapt to a variety of contexts and uses.

Graham — Mystery Powders Designer

Graham is a graduate student who wishes to create a family of assessment tasks to assess problem–solving skills in chemistry. The assessment involves an investigation that a student uses to identify the composition of a mixture of powders, using a series of tests such as adding re-agents, applying heat, and so on, and a complicated evaluation of the student's work products. The investigative steps repeat under the direction of the student, who proposes a final answer to finish the task. Graham uses the PDS to design the information needed in the screens that the students interact with, including the extensive feedback pages that indicate the history of previous steps and offer video clips. He describes the nature of the interactions and evaluation procedures. However, the PDS does not provide the affordances to create the screens themselves, the code workflow for presenting screens and controlling their progress, or the implementation of the evaluation procedures. Graham creates a prototype in Excel to display his investigation.

Patti — Tools Developer

Patti is a software developer who wishes to build a tool for creating complex assessments. She examines Graham's efforts with an eye toward abstracting the specific implementation to support the widest possible family of assessments that are similar in nature. Patti's mandate is to implement a *four-process architecture* (Almond et al., 2002) with the most transparency possible as an exemplar of how an adaptive assessment could be implemented. She draws up a system diagram that identifies the communication between the four processes of the delivery architecture and implements an overall system that allows for flexibility in defining elements, experiments, and rules.

Yolanda — Novice for Mystery Powder Investigations

Yolanda is a high-school teacher interested in creating Mystery Powders tasks for her Chemistry students. She maps out some experiments and results, resulting in a basic set of rules describing of six elements, six experiments, and all the resulting evidence. The authoring system has a special accommodation to build these kinds of Mystery assessments, and it asks Yolanda to describe the rules of her assessment. Yolanda enters the rules and enters pictures for the various outcomes of experiments. The authoring system then automatically produces the pages of the assessment, because the authoring system knows the rules that generate the pages. The authoring system offers to print the assessment on paper, or run it on computer.

Daquan (task author)

Daquan is a task developer for a commercial test publisher. He is assigned to create three Mystery Powders tasks for the item bank of a State X's computer-based science assessment program—one easy, one medium, and one hard. He works with a variation of the Wizard that the publisher's technical team has tuned to produce assessment objects that will run in the delivery system that State X uses.

We will discuss the Yolanda and Daquan use cases in greater depth within the context of our specific assessment and authoring system in section 5.2.

4.2 Choose Among PADI Exemplars Using Criteria

The next step after defining our specific goals was to choose an exemplar task within which to frame them. Given limited resources and several complex inquiry tasks that could be models for an authoring system, we chose among the following PADI-based projects based on the criteria mentioned earlier.

- Mystery Powders
- BioKIDs
- Floating Pencil
- Mystery Boxes
- FOSS
- NAEP

The final choice was to provide authoring for a family of ‘mystery’ tasks that are analogous to Mystery Powders. We selected Mystery Powders for several reasons. Mystery Powders tasks are interactive inquiry investigations. We have developed an operational working prototype for computer-based presentation and automated scoring of this familiar class of tasks in accordance with the *four-process delivery system architecture*. Full PADI *design patterns* and *templates* have been developed upon which to base an authoring wizard. Investigations developed from the wizard would be able to use the *template* structures and all of the delivery processes that have already been built for Mystery Powders.

4.3 Mockup Wizard in HTML

As a first step before implementation, an HTML mockup was created. This design was reviewed and several iterations of redesign followed.

4.4 Implement, Test (in future)

The implementation phase followed the design, and proceeded to the screen where the matrix of components and rules is displayed.

4.5 Publish

Given the prime directive of explicating our findings, we focused on this report in the final months of the grant extension, attempting to record a path for developing authoring systems that take advantage of the design objects that are created in the PADI Design System.

5.0 The Example: The Mystery Powders Assessment (MP-QTI)

5.1 Brief Overview

Mystery Powders is a typical hands-on lab experiment for middle-school science students. It asks students to perform a series of chemical and physical tests to determine the identity of a white powder. In PADI, it was used as an illustration of the *four-process delivery architecture*. To this end, the Mystery Powders investigation was reverse-engineered through analysis of the task and results. This information was put into the form of PADI *design patterns*, *templates*, and *task specifications*. These objects were subsequently used to build a computerized adaptive test that allowed participants to work through a series of Mystery Powders tasks (Seibert, Hamel, Haynie, Mislevy, & Bao 2006). This version used a flexilevel adaptive testing algorithm (Lord, 1971, 1980), which selects items based on the student's success answering the previous question. As such, a student would receive an easier question if the previous question was answered incorrectly and a harder question if the previous answer was correct. The Mystery Powders assessment built under PADI is called MP-QTI because it was delivered via the QTI standards.

The development of MP-QTI followed the PADI layers approach. During the *domain analysis*, the Mystery Powders tasks were identified as belonging to a larger class of tasks tapping hypothetico-deductive reasoning. In this case, Mystery Powders involves a finite solution space in which participants must think and reason with subject matter knowledge. For the MP-QTI, the processes of hypothetical reasoning, deductive reasoning, and strategic efficiency were chosen as the Focal KSAs.

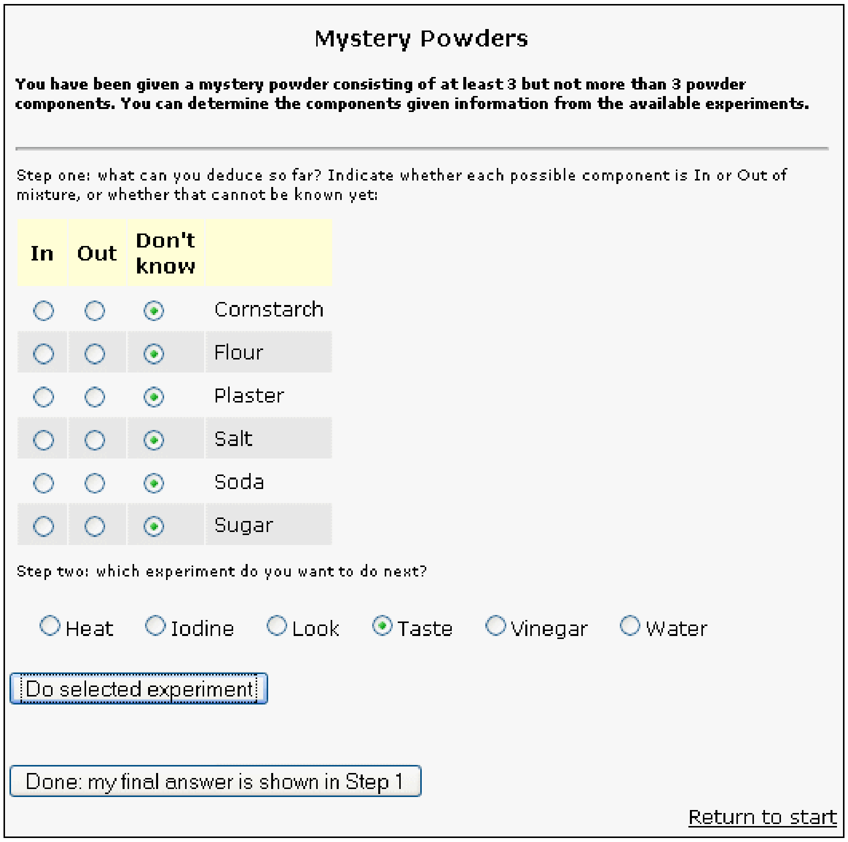
In the *domain modeling* layer, a *design pattern* for hypothetico-deductive tasks was created, with Mystery Powders as one example. A more specific *design pattern* for the Mystery Powders investigations was created by limiting tasks to those that involve determining the identity of unknown white powders using a set of chemical and physical tests. Participants could be scored on a variety of variables, including (but not limited to) correctness of their solution, the accuracy of their deductions after each test, and their choices of evidence. Subsequently, the Student, Task, and Evidence Models in the CAF were articulated for the Mystery Powders assessment.

The assessment was implemented using the *four-process delivery architecture* described previously. For the *assessment implementation*, tasks were created that followed the specification laid out in the CAF. For the demonstration version of the MP-QTI, 21 tasks that varied in difficulty were selected. The next section provides examples of the MP-QTI tasks from the student's perspective.

5.1.1 An MP-QTI Task from the Student's Perspective

This section demonstrates the MP-QTI tasks from the student's perspective as he or she interacts with the assessment. The screen shots and descriptions are taken directly from Seibert et al.'s 2006 technical report on the development of the MP-QTI. Figure 6 shows the initial presentation screen of MP-QTI.

Figure 6. Mystery Powders Presentation Screen



Mystery Powders

You have been given a mystery powder consisting of at least 3 but not more than 3 powder components. You can determine the components given information from the available experiments.

Step one: what can you deduce so far? Indicate whether each possible component is In or Out of mixture, or whether that cannot be known yet:

In	Out	Don't know	
○	○	●	Cornstarch
○	○	●	Flour
○	○	●	Plaster
○	○	●	Salt
○	○	●	Soda
○	○	●	Sugar

Step two: which experiment do you want to do next?

○ Heat ○ Iodine ○ Look ● Taste ○ Vinegar ○ Water

The screen shown in Figure 6 is made up of several blocks. First, a statement of the problem at the top declares the minimum and maximum number of components in the powder mixture along with an assurance that the mystery is determinable given the results of the available experiments[1]. The next part begins with a label for "Step One" and includes a solicitation of any conclusions the examinee can draw at the moment for each of the six potential powders. At this point, however, the examinee cannot draw any conclusions because no tests have been run; therefore, he or she would skip over this section on the initial screen. Below the Step One section is a Step Two section—a solicitation of the examinee's selection of the experiment to perform. At the bottom of the initial presentation screen are two buttons that give the examinee the choice of either proceeding with the selected experiment or finishing the task by declaring that the examinee has finished the task. Finally, the bottom right–hand corner of the screen shows a link to start over with a new powder mixture (new task).

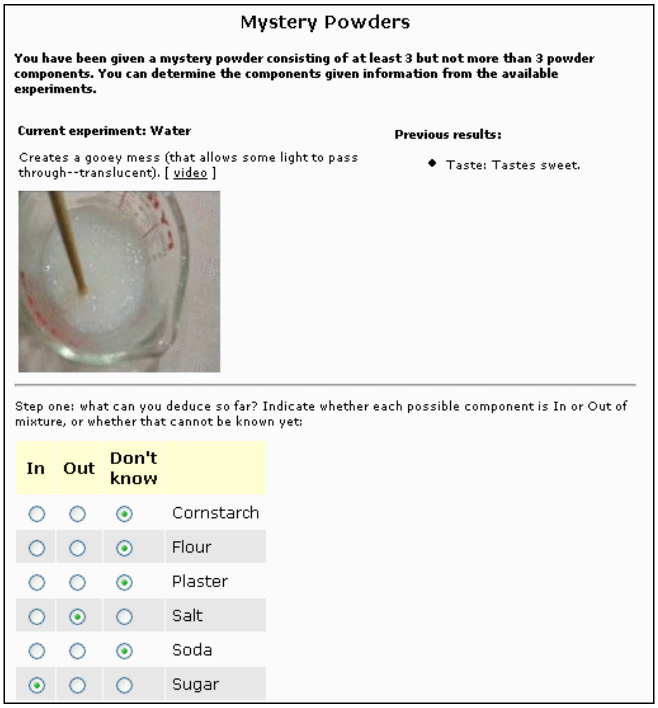
Following the initial presentation screen, the examinee will receive a series of screens providing the results to each selected test, soliciting the examinee's deductions following each test, requesting additional test selections, and allowing for the identification of a final solution. Figure 7 provides an example of these presentation screens. In the top section of Figure 7, a pictorial representation of the results for the water experiment is shown.

[1] Some powder combinations can be impossible to determine with the given experiments—they are indeterminable. In this MP-QTI implementation, we avoid giving indeterminable powders to examinees since such powders would add additional complexity to the already difficult tasks.

The water experiment, coupled with this particular powder combination, yielded a “gooey mess,” as described in the text accompanying the picture. An additional hyperlink provides a video of the experiment, showing water being stirred into the powder. In the upper right–hand part of the screen, the results of previous experiments are summarized. For this example, a previous taste experiment is summarized as “sweet”—indicating the presence of sugar and absence of salt. At the bottom of the screen (Step One) are six sets of radio buttons where the examinee can enter his or her deductions about the composition of the powder mixture.

The pictured display shows the examinee’s conclusions from the previous taste experiment, but none yet from the water experiment. Using the available evidence from the water experiment, the examinee could deduce that the “gooey mess” result implies cornstarch is in the powder combination and that plaster and flour are not in because these would mask the gooey result by resulting in a lumpy/muddy result and a lumpy/hard result, respectively. This recording of the examinee’s deductions (which comprise a Solution Matrix Work Product, as described in the *template*) is followed by another screen offering Step Two—the selection of an additional test to perform or choice to specify their deductions as final results.

Figure 7. Mystery Powders Presentation Screen 2



Mystery Powders

You have been given a mystery powder consisting of at least 3 but not more than 3 powder components. You can determine the components given information from the available experiments.

Current experiment: Water

Creates a gooey mess (that allows some light to pass through--translucent). [video]

Previous results:

- Taste: Tastes sweet.

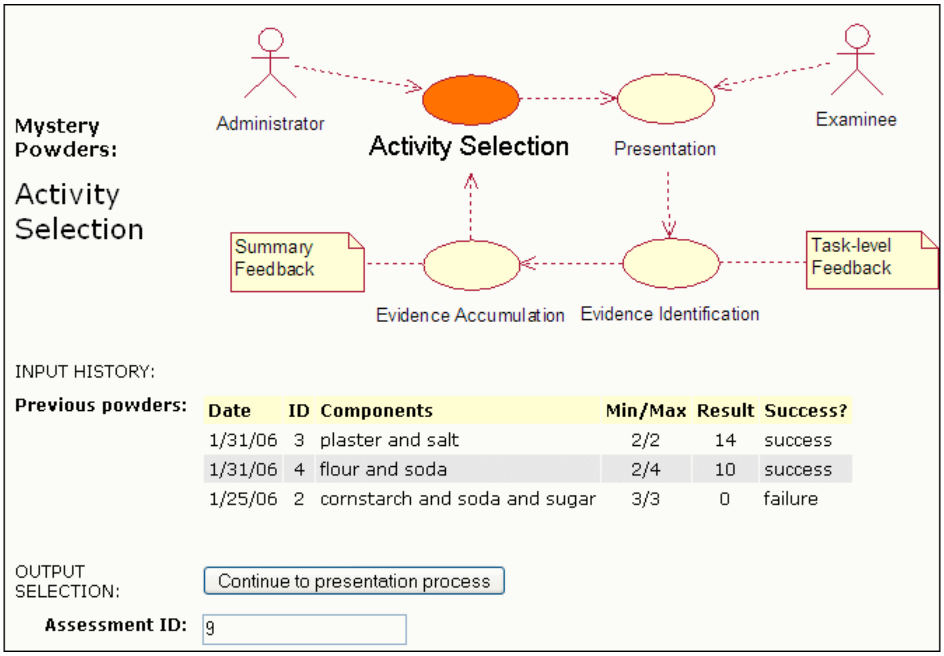
Step one: what can you deduce so far? Indicate whether each possible component is In or Out of mixture, or whether that cannot be known yet:

In	Out	Don't know	
○	○	●	Cornstarch
○	○	●	Flour
○	○	●	Plaster
○	●	○	Salt
○	○	●	Soda
●	○	○	Sugar

5.1.2 The MP-QTI Delivery System

The MP-QTI delivery system employs the *four-process architecture* approach (Almond et al., 2002). This section describes, through summary pages, how each of the processes act and interact during the *assessment implementation*. These summary pages and the processes they describe would not be seen by the examinee in actual testing; it is provided to make explicit the inputs, outputs, and interactions of the processes in the delivery system.

Figure 8. Summary Page for Activity Selection Process (Input Part)



Mystery Powders:

Activity Selection

INPUT HISTORY:

Previous powders:

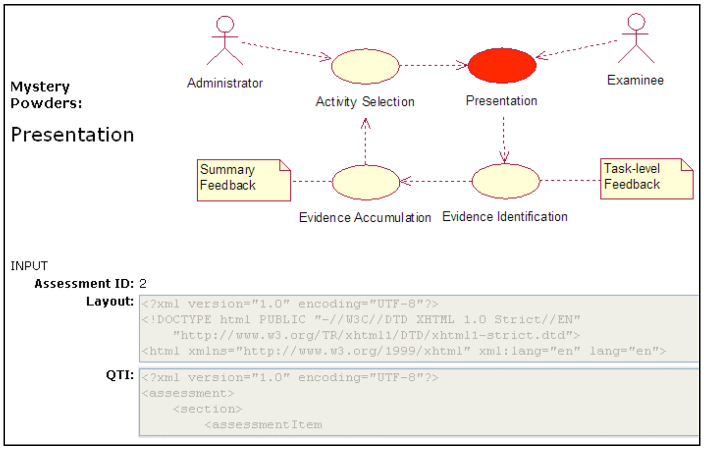
Date	ID	Components	Min/Max	Result	Success?
1/31/06	3	plaster and salt	2/2	14	success
1/31/06	4	flour and soda	2/4	10	success
1/25/06	2	cornstarch and soda and sugar	3/3	0	failure

OUTPUT SELECTION: [Continue to presentation process]

Assessment ID: 9

The input part of the summary page for the *activity selection process* is shown in Figure 8. It identifies the current process with a red oval and bold typeface within the original *four-process* diagram. Inputs to the selection process are the examinee's previous attempts, listed in the example in Figure 8 as three tasks (id numbers, powders, minimums, and maximums). The oldest performance listed on the figure (1/25/06) was a failure, and the more recent performances (both on 1/31/06) were successes. As a result, the next task selected will be slightly more difficult than most recent task (id number 3), as per the flexilevel algorithm.

Figure 9. Summary Page for Presentation Process (Input Part)



Mystery Powders:

Presentation

INPUT

Assessment ID: 2

Layout:

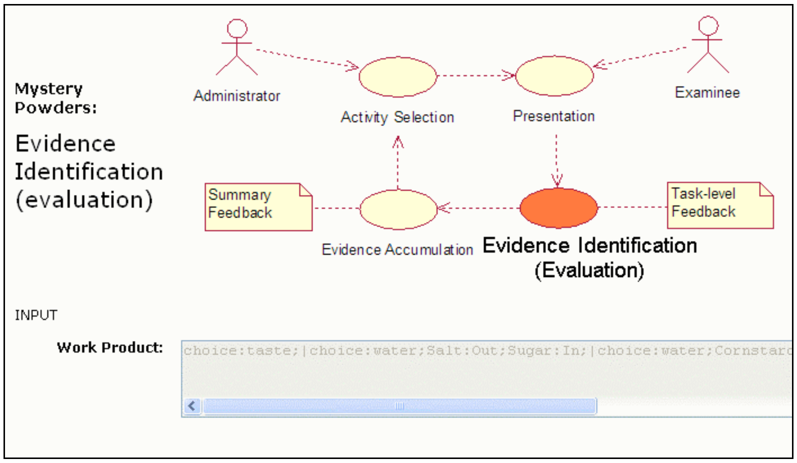
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```

QTI:

```
<?xml version="1.0" encoding="UTF-8"?>
<assessment>
    <section>
        <assessmentItem
```

The summary of the *presentation process* in Figure 9 shows all the inputs that come into the presentation, including the layout and QTI XML that was provided by the *activity selection process*. These inputs are shown on the summary page for information only and cannot be changed by a designer because they already have been used during the selection process.

Figure 10. Summary Page for Evidence Identification (Response Processing; Input Part)



Mystery Powders:

Evidence Identification (evaluation)

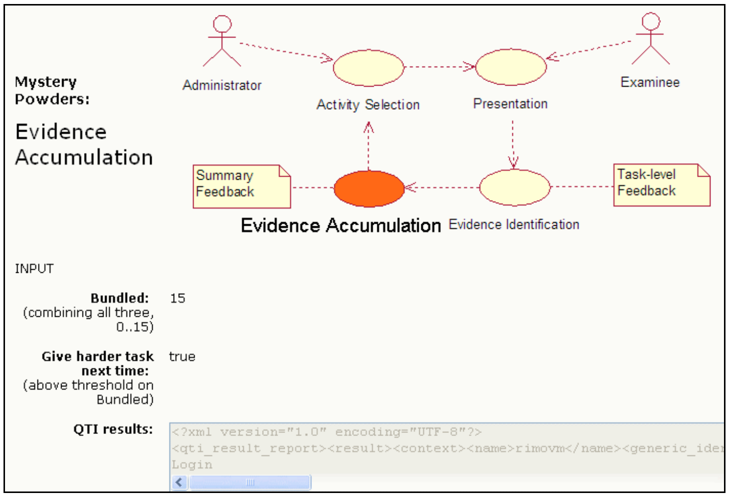
INPUT

Work Product:

```
choice:taste;|choice:water;Salt:Out;Sugar:In;|choice:water;Cornstarc
```

The summary of the *evidence identification process*, in Figure 10 shows the single input that came into this process, namely the string that summarizes the entire Work Product (selections of experiments and deductions following each test). This input, shown on the summary page, is for information only and cannot be changed because it already has been used during the process just completed.

Figure 11. Summary Page for Evidence Accumulation Process (Summary Scoring; Input Part)



Mystery Powders:

Evidence Accumulation

INPUT

Bundled: (combining all three, 0..15) 15

Give harder task next time: (above threshold on Bundled) true

QTI results:

```
<?xml version="1.0" encoding="UTF-8"?>
<qti_result_report><result><context><name>rimovm</name><generic_ider
Login
```

The summary of the *evidence accumulation process* in Figure 11 shows the inputs from the *evidence identification process*. These inputs are a final bundled Observable Variable score (15, in this example), determination of whether to administer a next, harder task (based on a threshold of '6' on the bundled observable variable), and some code for QTI results. These inputs are shown on the summary page for information only and cannot be changed because they already have been used during the *evidence identification process*.

5.2 Toward an Authoring Wizard

5.2.1 Overview of the form and purpose of the Authoring Wizard

As described in a previous paper on wizards in PADI (Hamel & Schank, 2006), the interview format of a wizard can help novice users focus on one thing at a time, facilitating the completion of a complex process. For our purposes, a wizard is essentially a factory for producing any number artifacts within a family of designs—the family of all the permutations that are allowed by making choices within the wizard. In the PADI Design System, a wizard helps create a *task template*, based on the prompts created beforehand by experts. The assessment expert who creates such a wizard helps others who use the wizard by narrowing the design choices, potentially leaving only a few questions to be answered by the person who uses the wizard.

Similarly, when authoring assessment tasks, we seek to facilitate the creation of tasks by embedding expertise within an authoring wizard. The first step is to use the information within a PADI *task specification*, where certain task information is complete and fixed, while other choices are left open. We present the open choices during the interview with the task author. The input from the task author is recorded within the wizard, and in the end, an assessment task is fully described. The fully described task can be called a task object. We prefer to capture this information in an extended QTI format, containing both QTI information about the item as well as some psychometric information about the Student and Measurement Models associated with the items.

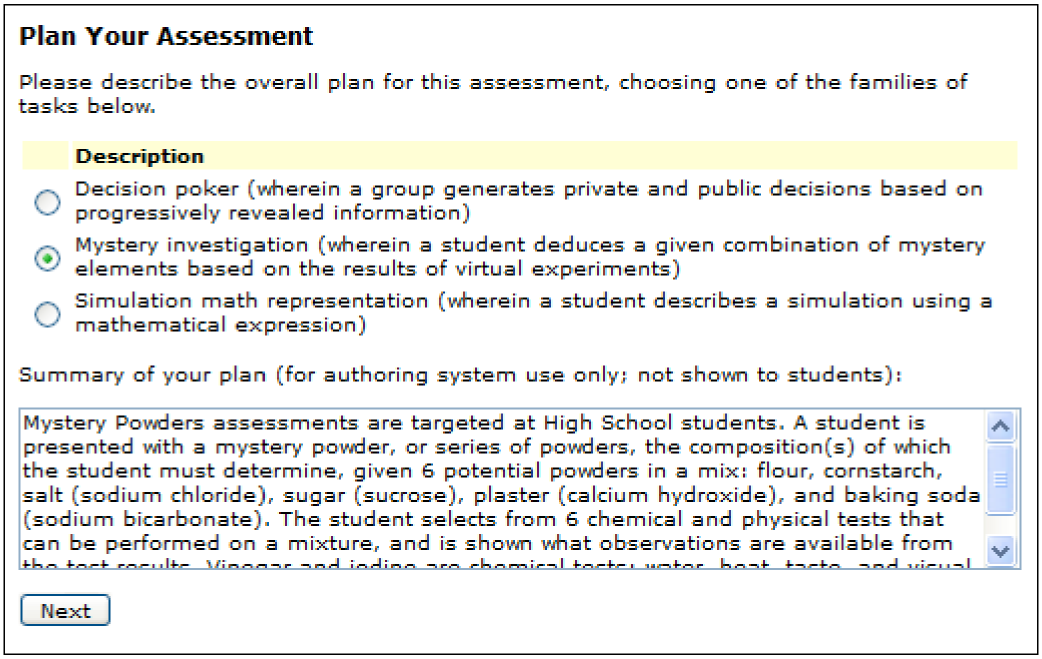
Because task objects must be compatible with the specific delivery systems in which they will be used, task objects are much more tailored than PADI *task templates*. The forms of task objects need to be determined beforehand by an assessment system designer (ideally compatible with QTI format). The task author working with the wizard has hidden from him/her the details and complexities of both the PADI *template* and the task object structures. Building this structure is accomplished by the collaboration of a wizard author and a master task designer. This work is motivated by the Graham use-case. Note that we want wizard authoring structures to be as re-usable as possible, but this is constrained by the fact that the output they produce is specific to a given delivery system.

5.2.2 Sample Screens—Scenario 1

Consider the use-case of Yolanda, the high-school teacher interested in a mystery task. For ease in creating examples, let us say that her focus will be in high-school chemistry, such that our existing example of Mystery Powders will suffice for her.

In Figure 12, Yolanda begins the wizard by selecting the kind of task she will author. She selects the second type, a Mystery investigation task, from the choices available in the authoring system. (Within this phase of the research project, we have designed, but not implemented, only the Mystery investigation option; the other options are listed only as examples of complex assessment tasks that might be provided by a wizard system.)

Figure 12. Plan Your Assessment



Plan Your Assessment

Please describe the overall plan for this assessment, choosing one of the families of tasks below.

	Description
○	Decision poker (wherein a group generates private and public decisions based on progressively revealed information)
●	Mystery investigation (wherein a student deduces a given combination of mystery elements based on the results of virtual experiments)
○	Simulation math representation (wherein a student describes a simulation using a mathematical expression)

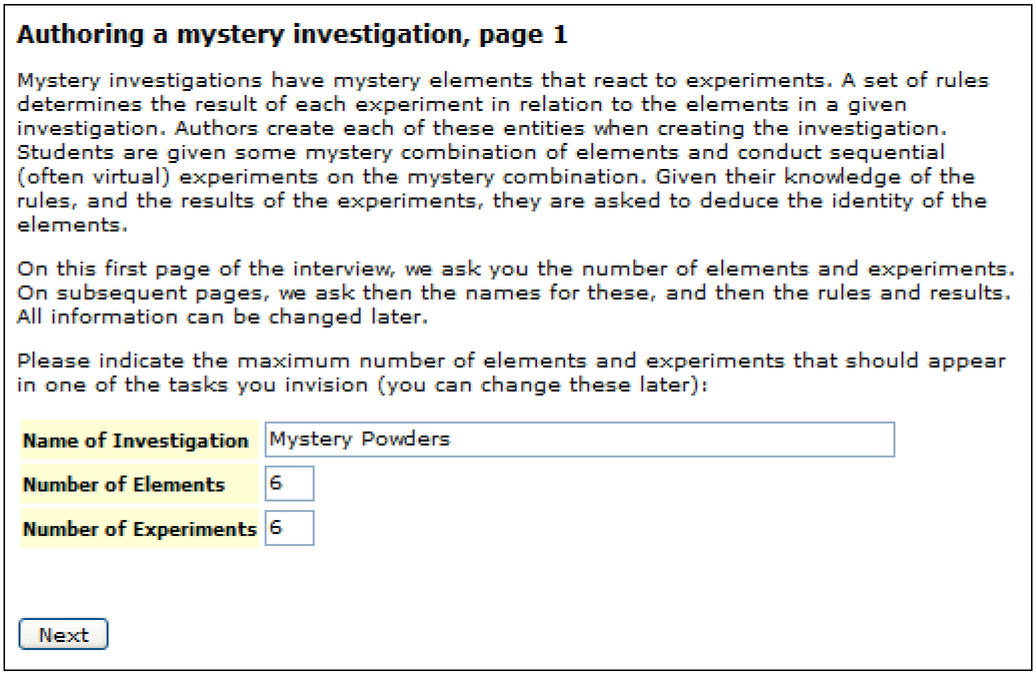
Summary of your plan (for authoring system use only; not shown to students):

Mystery Powders assessments are targeted at High School students. A student is presented with a mystery powder, or series of powders, the composition(s) of which the student must determine, given 6 potential powders in a mix: flour, cornstarch, salt (sodium chloride), sugar (sucrose), plaster (calcium hydroxide), and baking soda (sodium bicarbonate). The student selects from 6 chemical and physical tests that can be performed on a mixture, and is shown what observations are available from the test results. Vinegar and iodine are chemical tests; water, heat, taste, and visual

In Figure 12, we have simulated the summary Yolanda might enter. When Yolanda clicks “Next,” her choice tells the authoring system to pull in information from the PADI *task specification* associated with Mystery Investigations. The linkage between the specification and the choice is flexible. The specification can be changed, and those changes would flow down into the authoring system dynamically because the authoring system will pull down (e.g., HTTP download) the specification each time it needs to begin a wizard. This linkage, up to and including a URL to the relevant object in a PADI Design System, could even be made more evident and editable in the interface, but we wish to hide complexity in this situation. Yolanda is depending on the authoring system implementers to have linked to the best *task specification* possible.

Next, Yolanda would see Figure 13, where she is asked to name this task and indicate two numbers.

Figure 13. Name Your Investigation



Authoring a mystery investigation, page 1

Mystery investigations have mystery elements that react to experiments. A set of rules determines the result of each experiment in relation to the elements in a given investigation. Authors create each of these entities when creating the investigation. Students are given some mystery combination of elements and conduct sequential (often virtual) experiments on the mystery combination. Given their knowledge of the rules, and the results of the experiments, they are asked to deduce the identity of the elements.

On this first page of the interview, we ask you the number of elements and experiments. On subsequent pages, we ask then the names for these, and then the rules and results. All information can be changed later.

Please indicate the maximum number of elements and experiments that should appear in one of the tasks you invision (you can change these later):

Name of Investigation	Mystery Powders
Number of Elements	6
Number of Experiments	6

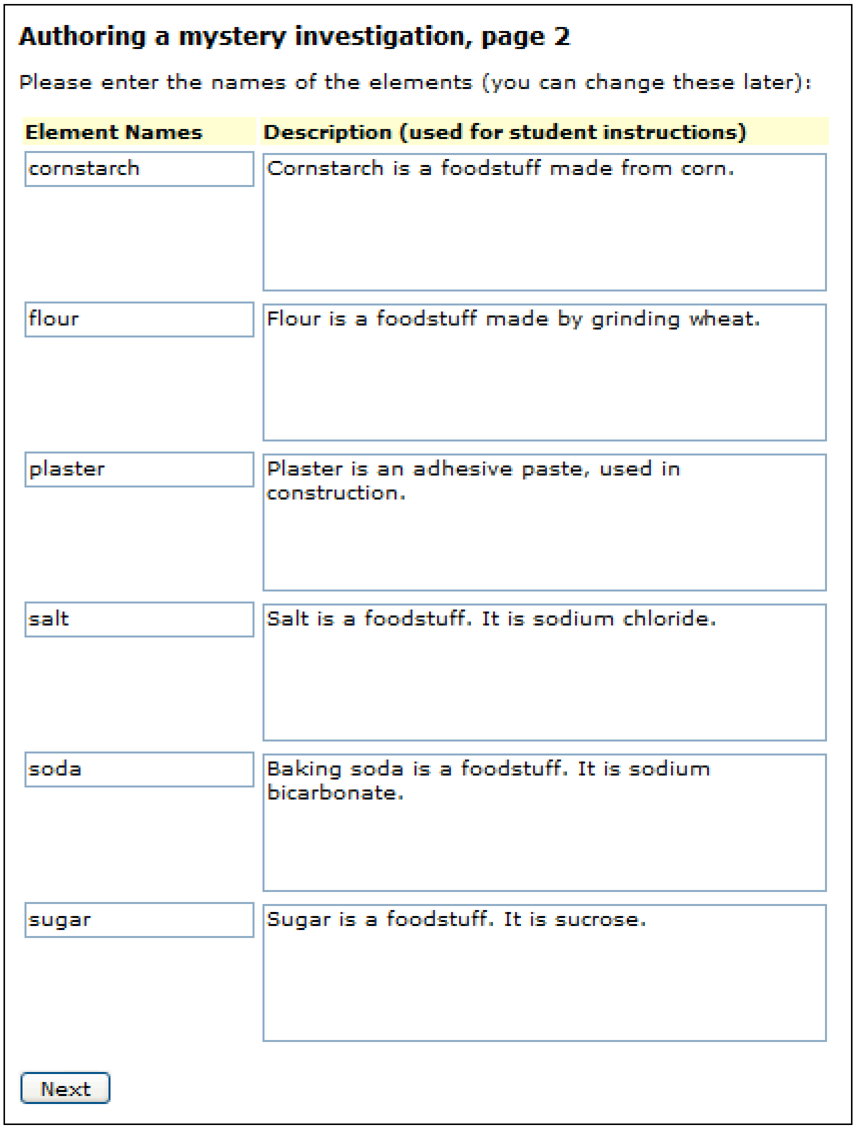
There are two sets of components, as explained in the text of Figure 13: the experiments that the student performs and the elements of the mystery that, when deduced, will solve the mystery. The author is asked for these numbers so that the authoring system can offer a screen to enter all the elements at once, and then all the experiments at once. However, other designs are possible, and even in this mockup, additional elements and experiments can be added later or removed. Behind the scenes, the authoring system can prepare a matrix of permutations for the various combinations of elements.

The *task specification* upon which this authoring wizard is based will have some constraints on the maximum number of elements and experiments, based on expert experience. Similarly, a minimum number would be specified to make the task non-trivial. The *task specification* would have these constraints within PADI Task Model Variables, and the authoring system must parse and enforce such constraints.

Yolanda's input is saved in a database that is relatively customized for mystery investigations and is focused on creating QTI+ representations, the same kind of representations used in the Mystery Powders demonstration software.

When Yolanda enters her expected totals for both components and clicks "Next," she sees the screen of Figure 14.

Figure 14. Enter Element Names



Authoring a mystery investigation, page 2

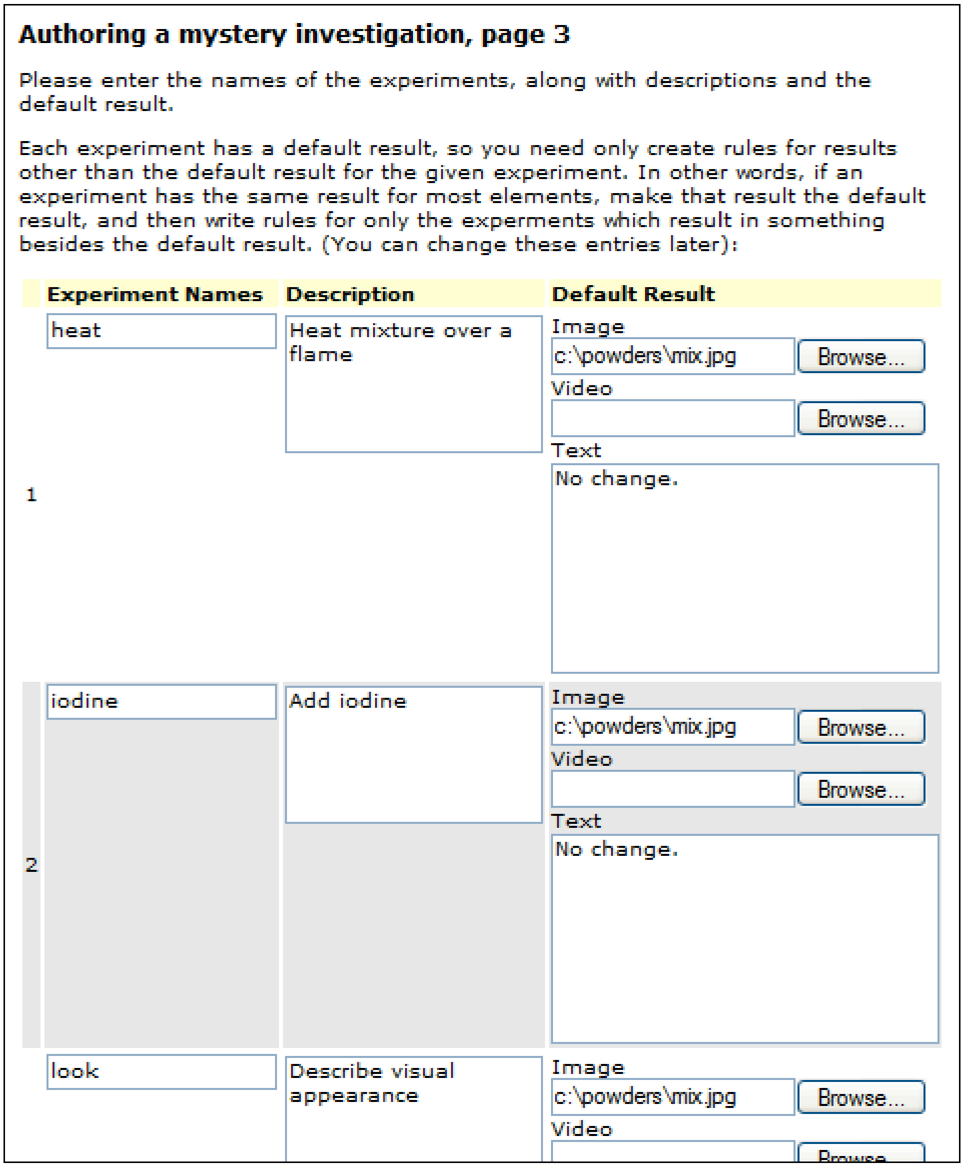
Please enter the names of the elements (you can change these later):

Element Names	Description (used for student instructions)
cornstarch	Cornstarch is a foodstuff made from corn.
flour	Flour is a foodstuff made by grinding wheat.
plaster	Plaster is an adhesive paste, used in construction.
salt	Salt is a foodstuff. It is sodium chloride.
soda	Baking soda is a foodstuff. It is sodium bicarbonate.
sugar	Sugar is a foodstuff. It is sucrose.

Figure 14 shows how Yolanda might name and describe the elements in her mystery investigation. In this example, all of the elements are powders that may be combined into a mystery mixture.

When Yolanda finishes this screen and clicks "Next," she sees the screen of Figure 15.

Figure 15. Enter Experiments



Authoring a mystery investigation, page 3

Please enter the names of the experiments, along with descriptions and the default result.

Each experiment has a default result, so you need only create rules for results other than the default result for the given experiment. In other words, if an experiment has the same result for most elements, make that result the default result, and then write rules for only the experments which result in something besides the default result. (You can change these entries later):

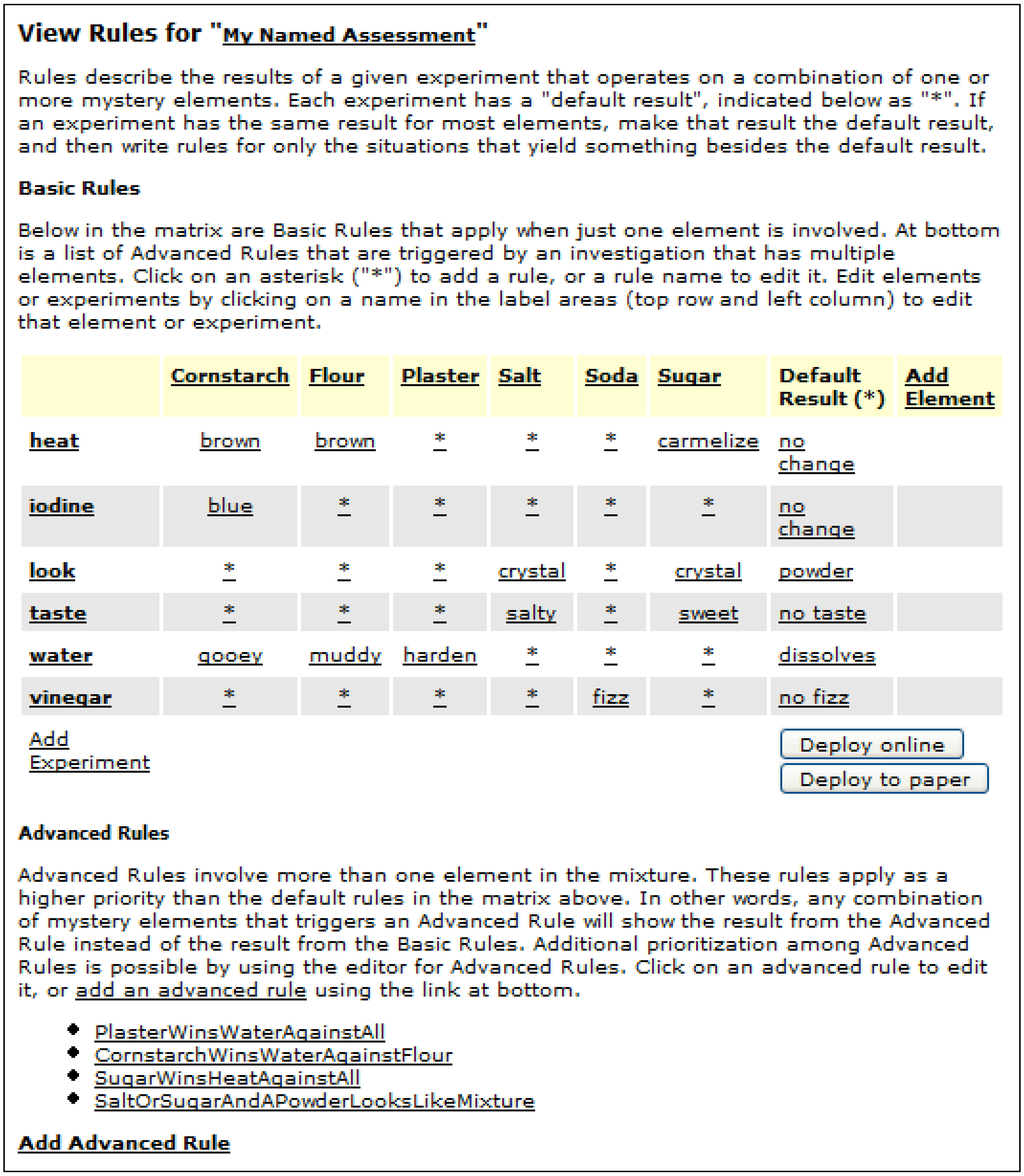
	Experiment Names	Description	Default Result
1	heat	Heat mixture over a flame	Image c:\powders\mix.jpg Browse... Video Browse... Text No change.
2	iodine	Add iodine	Image c:\powders\mix.jpg Browse... Video Browse... Text No change.
	look	Describe visual appearance	Image c:\powders\mix.jpg Browse... Video Browse

The screen for Figure 15 extends below the image provided and includes fields for the specified number of experiments (in our example, 6) as well as the familiar "Next" button. The entry of experiments is straight forward, like the entry of elements—a name and a description—but this is also the first introduction to the results that are delivered by the experiments. This screen provides optional input for a default result, the result that is most common when an experiment is undertaken. Typically this default result is the null hypothesis—nothing happens. The input area for the default result includes a mandatory description in text and two optional descriptions in image and video.

Behind the scenes, as soon as the system knows about the experiments and the first inkling of results, it can begin a brute–force examination of the universe of tasks to determine whether each permutation has a unique solution, how each possible sequence of experiments is more or less efficient (e.g., what is the best experiment to do first?), how many experiments are minimally necessary to determine the correct answer for a given task (to provide one measure of task difficulty), etc.

When Yolanda finishes entering the experiments, and optionally their default results, she sees the summary screen of Figure 16.

Figure 16. Rules Matrix



View Rules for "My Named Assessment"

Rules describe the results of a given experiment that operates on a combination of one or more mystery elements. Each experiment has a "default result", indicated below as "*". If an experiment has the same result for most elements, make that result the default result, and then write rules for only the situations that yield something besides the default result.

Basic Rules

Below in the matrix are Basic Rules that apply when just one element is involved. At bottom is a list of Advanced Rules that are triggered by an investigation that has multiple elements. Click on an asterisk ("*") to add a rule, or a rule name to edit it. Edit elements or experiments by clicking on a name in the label areas (top row and left column) to edit that element or experiment.

	Cornstarch	Flour	Plaster	Salt	Soda	Sugar	Default Result (*)	Add Element
heat	brown	brown	*	*	*	carmelize	no change	
iodine	blue	*	*	*	*	*	no change	
look	*	*	*	crystal	*	crystal	powder	
taste	*	*	*	salty	*	sweet	no taste	
water	gooey	muddy	harden	*	*	*	dissolves	
vinegar	*	*	*	*	fizz	*	no fizz	

Advanced Rules

Advanced Rules involve more than one element in the mixture. These rules apply as a higher priority than the default rules in the matrix above. In other words, any combination of mystery elements that triggers an Advanced Rule will show the result from the Advanced Rule instead of the result from the Basic Rules. Additional prioritization among Advanced Rules is possible by using the editor for Advanced Rules. Click on an advanced rule to edit it, or add an advanced rule using the link at bottom.

- PlasterWinsWaterAgainstAll
- CornstarchWinsWaterAgainstFlour
- SugarWinsHeatAgainstAll
- SaltOrSugarAndAPowderLooksLikeMixture

Add Advanced Rule

Figure 16 gives Yolanda an overview of her task, including its elements, experiments, and rules. Across the top of the matrix, the elements she entered are displayed as column names. There are two additional columns on the right, one for the default result that Yolanda entered, and one for adding a new element. Each existing element can be edited by clicking on its name in the column header.

Each experiment that Yolanda entered is represented by a row, with an additional row for adding a new experiment. Each experiment can be edited by clicking on the row name in the leftmost column.

The cells of the matrix contain the basic rules, the rules that apply when there is only one element in the mystery mixture. An asterisk in a cell indicates that the default result applies. A name in a cell indicates that a default–overriding rule applies, with the given name. For example, in the upper– left corner of the matrix, when cornstarch undergoes a heat experiment, it turns brown.

At the bottom of the screen, a list of advanced rules is provided. For situations where a result depends on the combination of elements present in the mixture, advanced rules allow a means to determine the result. For example, the rule "PlasterWinsWaterAgainstAll" is a sample advanced rule wherein the result of adding water to any powder mixture (within our universe of element mixtures) that contains plaster will turn that powder into a solid substance suitable for patching walls.

For deployment, two buttons on the right offer to take the finished task and run it on the computer or print it to paper.

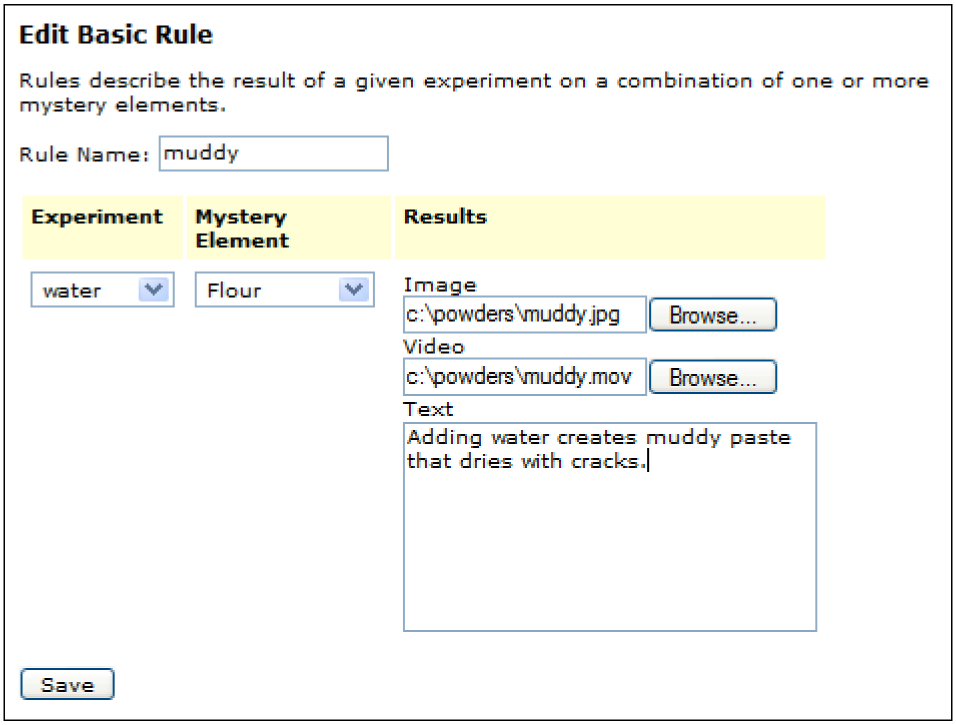
There is no "Next" button on this screen. It is here that Yolanda finishes authoring her tasks, albeit with detours to editing screens for rule creation and other editing, returning here to the summary matrix after each edit. After each edit, the system would need to rerun internal checks to identify conflicts and determine which, if any, tasks had unique answers and were most difficult (by measures such as minimum number of experiments, etc.)

To show more detail of the editing related to this screen, we'll describe the following links found on this screen in this order:

- edit a basic rule
- edit an advanced rule

First, consider the screen for editing a basic rule in Figure 17.

Figure 17. Edit Basic Rule



Edit Basic Rule

Rules describe the result of a given experiment on a combination of one or more mystery elements.

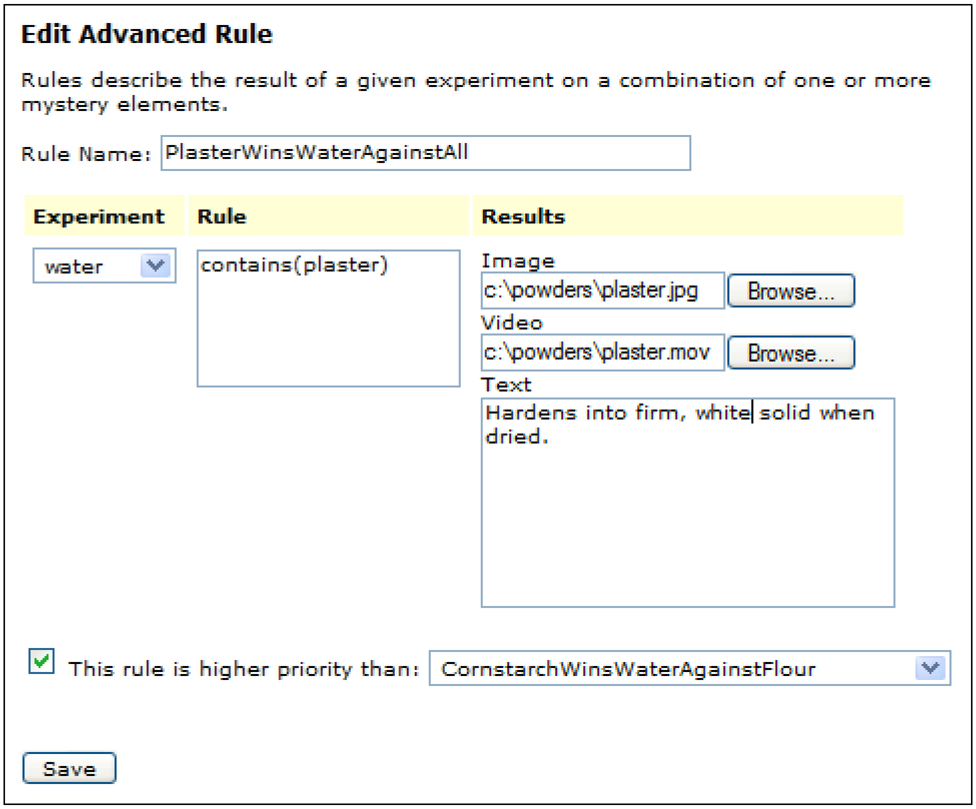
Rule Name: muddy

Experiment	Mystery Element	Results
water	Flour	Image c:\powders\muddy.jpg Video c:\powders\muddy.mov Text Adding water creates muddy paste that dries with cracks.

Figure 17 shows an example basic rule named “muddy,” the result of adding water to a powder with only one element—flour. There is a single experiment and a single element that determine when this rule will apply. The result yielded by the experiment is described with text and optional image and video.

An advanced rule is shown next in Figure 18.

Figure 18. Edit Advanced Rule



Edit Advanced Rule

Rules describe the result of a given experiment on a combination of one or more mystery elements.

Rule Name: PlasterWinsWaterAgainstAll

Experiment	Rule	Results
water	contains(plaster)	Image c:\powders\plaster.jpg Video c:\powders\plaster.mov Text Hardens into firm, white solid when dried.

☑ This rule is higher priority than: CornstarchWinsWaterAgainstFlour

Figure 18 shows input for a complex rule, including the experiment and a text field labeled “Rule” and filled here with a sample script reading “contains(plaster).” The intent of this example is to imply that this rule should fire for any mixture that contains plaster. A script parser is a flexible but somewhat challenging feature. As an alternative to scripting, it might be possible to have a matrix of check–boxes to indicate the presence or absence (or “don’t care”) of all elements to describe when the rule fires. The particular implementation strategy is a detail left up to future developers.

At the bottom of this screen, a check–box and drop-down menu provide for prioritization of advanced rules. In this example, the plaster rule has a higher priority than the cornstarch rule. In other words, a mixture with both plaster and cornstarch will yield the result described by the plaster rule, ignoring the cornstarch rule. Likewise, the cornstarch rule “CornstarchWinsAgainstFlour,” implies that there is a prioritization there also, such that a mixture with both cornstarch and flour (but not plaster) will yield the result for cornstarch.

The rules can be parsed by the system to check for and flag conflicts, such as two rules that both claim to take priority for a given experiment. Such rules would need to be colored red or otherwise brought to the attention of the author.

Complex chains of rules might be difficult to understand at a glance, so we can imagine a screen representation using a flow chart like the one in Figure 19.

Figure 19. Flowchart of Relationship Among Advanced Rules

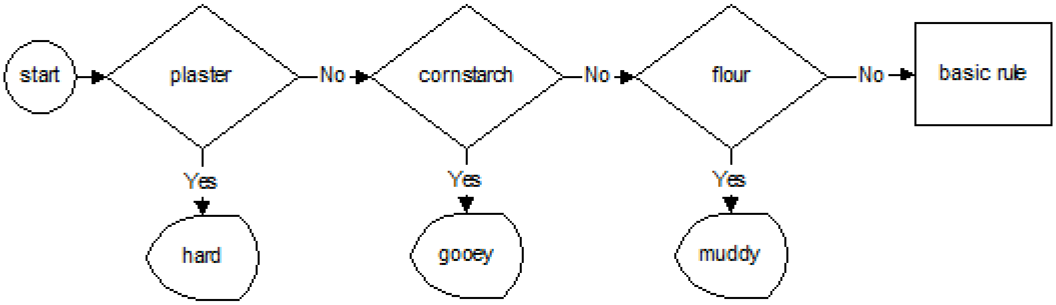


Figure 19 shows a flow of control such that that the existence of plaster trumps other choices, while the existence of cornstarch trumps flour, etc. The authoring system could conceivably generate this diagram from the rules entered or could even provide some kind of visual editing that created rules by creating flowcharts (again, an implementation option). In any case, the rules must be captured and applied in an internally consistent manner so that conflicts can be identified and fixed by the author.

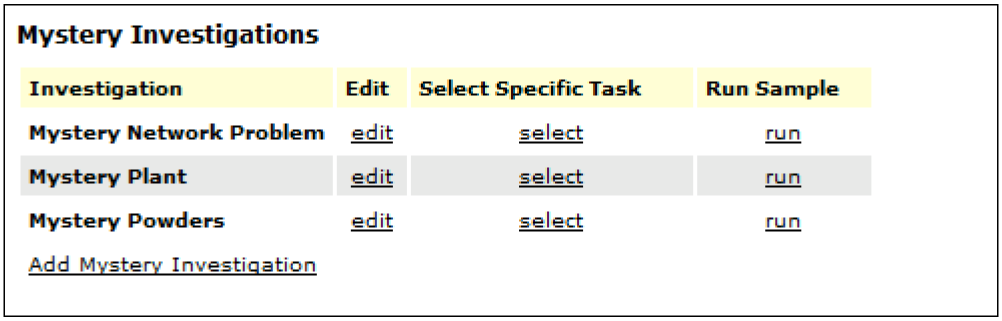
Additional authoring pages could offer visual customizations and other features. We have largely focused on the pages needed to set up the key functionality.

5.2.2 Sample Screens—Scenario 2

Taking up the Daquan use-case, we have an author who must add three tasks at three different levels of difficulty to a pool of assessment items,. Daquan is using an authoring system at a commercial test publisher, where elements, experiments and rules have already been described. It is Daquan's task to create tasks based on this foundation.

First, he reviews the list of investigations (families of related mystery tasks) already available, as shown in Figure 20.

Figure 20. List of Investigations (Families of Tasks)



Mystery Investigations

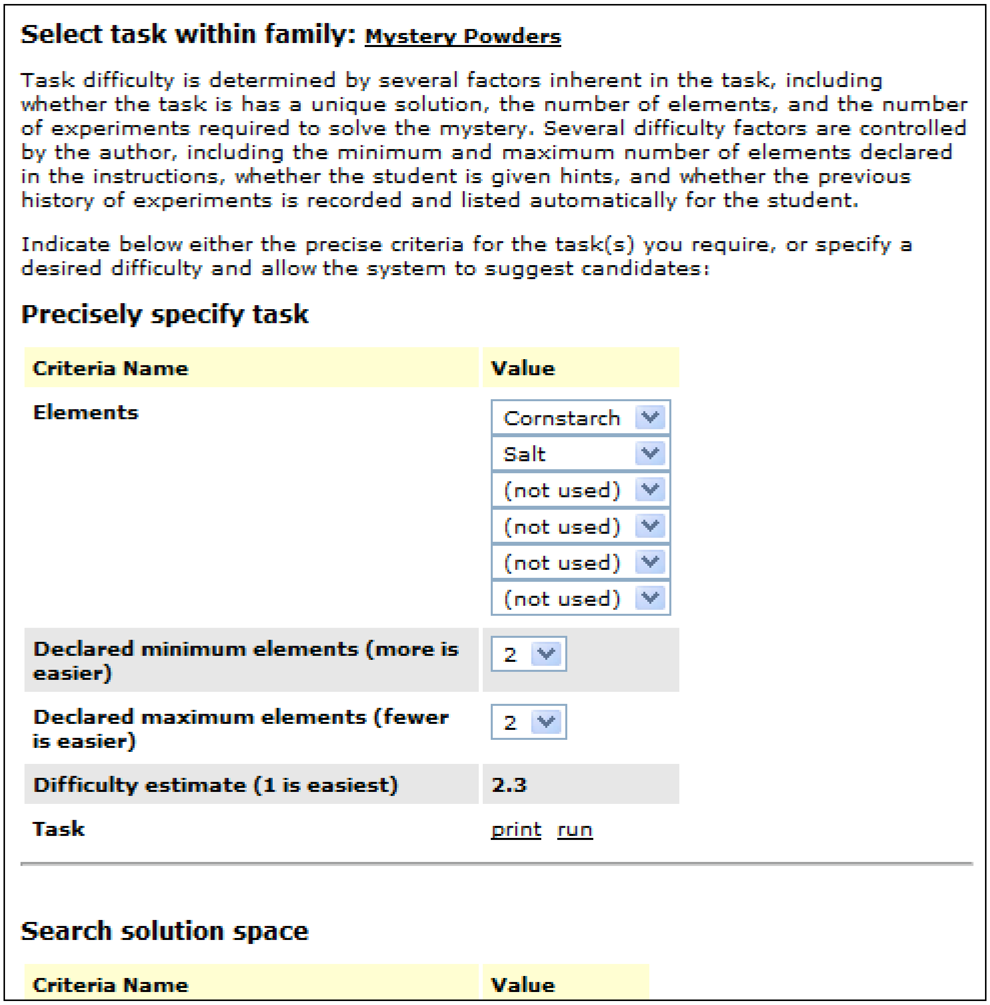
Investigation	Edit	Select Specific Task	Run Sample
Mystery Network Problem	edit	select	run
Mystery Plant	edit	select	run
Mystery Powders	edit	select	run

Add Mystery Investigation

Daquan has the opportunity here to edit an existing investigation, see a sample task, add a brand new investigation (which would show the first screen that Yolanda saw), or select a specific task from within the family.

Daquan needs three tasks from the Mystery Powders family, so he clicks the link to select Mystery Powders and proceeds to the screen of Figure 21.

Figure 21. Selecting Tasks by Difficulty, Part 1



Select task within family: Mystery Powders

Task difficulty is determined by several factors inherent in the task, including whether the task is has a unique solution, the number of elements, and the number of experiments required to solve the mystery. Several difficulty factors are controlled by the author, including the minimum and maximum number of elements declared in the instructions, whether the student is given hints, and whether the previous history of experiments is recorded and listed automatically for the student.

Indicate below either the precise criteria for the task(s) you require, or specify a desired difficulty and allow the system to suggest candidates:

Precisely specify task

Criteria Name	Value
Elements	Cornstarch Salt (not used) (not used) (not used) (not used)
Declared minimum elements (more is easier)	2
Declared maximum elements (fewer is easier)	2
Difficulty estimate (1 is easiest)	2.3
Task	print run

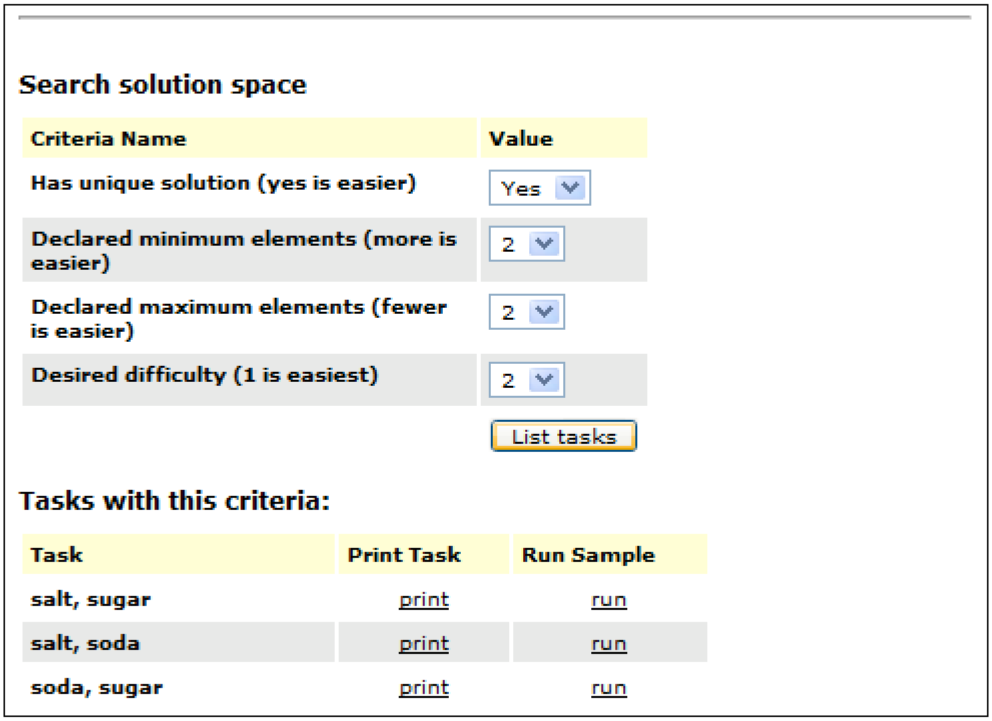
Search solution space

Criteria Name	Value

Figure 21 explains to Daquan that task difficulty is determined by several criteria, some criteria inherent to the task and other criteria peripheral to the task, such as how much support the student is given.

The top part of the screen allows Daquan to precisely specify the task, including the powder elements. As a result of his specification, the system makes an estimate of task difficulty based on internal algorithms which may have the benefit of calibration from trials, or be based on developer assumptions such as “more elements mean more difficulty.” The screen continues in Figure 22.

Figure 22. Selecting Tasks by Difficulty, Part 2



Search solution space

Criteria Name	Value
Has unique solution (yes is easier)	Yes
Declared minimum elements (more is easier)	2
Declared maximum elements (fewer is easier)	2
Desired difficulty (1 is easiest)	2

Tasks with this criteria:

Task	Print Task	Run Sample
salt, sugar	print	run
salt, soda	print	run
soda, sugar	print	run

In Figure 22, the system allows Daquan to survey the solution space by entering a criteria of desired difficulty and seeing what tasks fit that description when Daquan clicks "List Tasks."

In the list displayed in Figure 22, there are three tasks, each with two elements (as per criteria), and with relatively low difficulty. Daquan can select any one of these to try a sample run, or to print out.

The choices Daquan makes when using the wizard inform the contents of the task object as they have been defined more generically in the PADI *template*. Behind the scenes, the authoring system has used internal algorithms, including an exhaustive mapping of the solutions of all tasks in the family, to estimate difficulty and index individual tasks by that difficulty. When Daquan clicks to run or print a task, its internal format is translated into QTI to be sent to a rendering process for running on a computer or printing.

6.0 Discussion

6.1 Benefits of this Specific Example

The Mystery Powders assessment example outlined in this paper has several inherent benefits. These benefits include the ability to:

- **Generalize to other adaptive "Mystery" assessments.** Other inquiry assessments are based on some unknown, or mystery, than needs to be identified or solved. The Mystery Powders assessment can be used to create different types of mystery tasks.

- **Reuse Mystery Powders implementation as alpha test.** The existing implementation of Mystery Powders (http://cltnet.org/powders/login.jsf) accommodates the input of QTI+ information that is rendered as an assessment. As an alpha test of the authoring system, an assessment will be created that closely mimics the number of components and other details of the current Mystery Powders implementation. The resulting QTI+, when interposed in the current Mystery Powders implementation, should provide the same experience as that existing assessment.
- **Serve as an exemplar**. Creating an authoring wizard for a working computer-based interactive assessment, cast in terms of the *four-process delivery architecture* and using QTI data structures, provides an exemplar for applying the approach to other task types and settings.

6.2 General Benefits of the Approach

The example can help improve aspects of current Mystery Powders activities, such as the hard-coded approach to task difficulty and experiment-sequence efficiency. For example, certain experiments (such as iodine revealing cornstarch) will only reveal information about one of the possible ingredients and are therefore relatively weak as a starting experiment. The system can determine the strength and weakness of experiments by exhaustively attempting all combinations and recording results. The strength of experiments to reveal information is associated with one of the Student Model Variables (the student's efficiency in selecting the most powerful experiments, given the stage of the investigation). See Seibert et al. (2006) for further details.

There are global benefits to assessment design and delivery inherent in the approach outlined in this paper. These benefits include:

- **Addressing the question "How can I use PADI?," without resorting to the "Blueprint But Not House" analogy.** An authoring system will finally give us a better answer to practitioners who want immediate and relatively simple access to PADI structures and principles. As the use-cases demonstrate, an authoring system built to draw on PADI structures and principles can be used by different parties involved in assessment, from a classroom teacher to a commercial test constructor.

- **Demonstrating how PADI design information can flow into an authoring system.** The PADI Design System can produce an assessment design that is later ignored or misunderstood during subsequent steps that lead up to a real assessments deployment. Providing a linked Authoring System provides more constraints, and thereby a more easily validated implementation of a given design. By linking the Authoring System with PADI structures and principles, the assessment argument is carried through to the implementation and delivery stages of the assessment process.

- **Improving the PADI Design System as authoring issues become evident.** As the Authoring System is implemented and used, we will see how improvements to the Design System will make an assessment design flow more smoothly into an actual assessment. This type of iterative feedback and improvement at every level is essential to maintaining the assessment argument through the layers of the assessment enterprise.

References

Almond, R. G., Steinberg, L. S., & Mislevy, R. J. (2002). Enhancing the design and delivery of assessment systems: A four-process architecture. *Journal of Technology, Learning, and Assessment, 1(5)*. Available at http://www.bc.edu/research/intasc/jtla/journal/v1n5.shtml

Bachman, L. F. (2003). Building and supporting a case for test use. *Language Assessment Quarterly, 2,* 1-34.

Baxter, G. P., Elder, A. D., & Glaser, R. (1995). Cognitive analysis of a science performance assessment. *CSE Technical Report 398.* Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST), Center for Studies in Education, UCLA.

Hamel, L., & Schank, P. (2006). *A Wizard for PADI assessment design (PADI Technical Report 11).* Menlo Park, CA: SRI International.

Lord, F. M. (1971). The self-scoring flexi-level test. *Journal of Educational Measurement, 8,* 147-151.

Lord, F. M. (1980). *Application of item response theory to practical testing problems*. Hillsdale, NJ: Erlbaum.

Luecht, R. M. (2002). From design to delivery: Engineering the mass production of complex performance assessments. Paper presented at the Annual Meeting of the National Council on Measurement in Education, New Orleans, LA.

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

Mislevy, R. J. (2003). Substance and structure in assessment arguments. *Law, Probability, and Risk, 2,* 237-258.

Mislevy, R. J., Behrens, J. T., Bennet, R. E., et al. (2007). On the roles of external knowledge representations in assessment design (*CSE Technical Report).* Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST), Center for Studies in Education, UCLA.

Mislevy, R. J., & Haertel, G. (2006). Implications for evidence-centered design for educational assessment. *Educational Measurement: Issues and Practice, 4,* 6-20.

Mislevy, R. J., & Riconscente, M.M. (2006). Evidence-centered assessment design: Layers, structures, and terminology. In S. Downing & T. Haladyna (Eds.), *Handbook of Test Development* (pp. 61-90). Mahwah, NJ: Erlbaum.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2002). On the roles of task model variables in assessment design. In S. Irvine & P. Kyllonen (Eds.), *Generating items for cognitive tests: Theory and practice* (pp. 97-128). Hillsdale, NJ: Erlbaum.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2003). On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives, 1,* 3-67.

Riconscente, M. M., Mislevy, R. J., & Hamel, L. (2005). *An introduction to PADI task templates.* (PADI Technical Report 3). Menlo Park, CA: SRI International.

Seibert, G., Hamel, L., Haynie, K., Mislevy, R., & Bao, H. (2006). *Mystery powders: An application of the PADI design system using the four-process delivery system* (PADI Technical Report 15). Menlo Park, CA: SRI International.

Toulmin, S. E. (1958). *The uses of argument.* Cambridge: Cambridge University Press.

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*