A Framework for Structuring Learning Assessment in a Massively Multiplayer Online Educational Game: Experiment Centered Design

Shawn Conrad, Jody Clarke-Midura, Eric Klopfer
The Education Arcade, Massachusetts Institute of Technology
20 Ames Street, Cambridge, MA 02139
Telephone: 617-253-2025
Email: shawnrad@mit.edu, jodycm@mit.edu, klopfer@mit.edu

A Framework for Structuring Learning Assessment in a Massively Multiplayer Online Educational Game: Experiment Centered Design

ABSTRACT – Educational games offer an opportunity to engage and inspire students to take interest in science, technology, engineering, and mathematical (STEM) subjects. Unobtrusive learning assessment techniques coupled with machine learning algorithms can be utilized to record students' in-game actions and formulate a model of the students' knowledge without interrupting the students' play. This paper introduces "Experiment Centered Assessment Design" (XCD), a framework for structuring a learning assessment feedback loop. XCD builds on the "Evidence Centered Assessment Design" (ECD) approach, which uses tasks to elicit evidence about students and their learning. XCD defines every task as an experiment in the scientific method, where an experiment maps a test of factors to observable outcomes. This XCD framework was applied to prototype quests in a massively multiplayer online (MMO) educational game. Future work would build upon the XCD framework and use machine learning techniques to provide feedback to students, teachers, and researchers.

Key words: Experiment Centered Assessment Design; Evidence Centered Assessment Design; Learning Assessment; Massively Multiplayer Online Role Playing Game (MMORPG);

I. INTRODUCTION

Open-world games like massively multiplayer online role playing games (MMORPGs) encourage exploration and experimentation. In these environments, learning is situated in problem spaces that involve hypothesizing, probing, observing, reflecting, and recycling these steps (Gee, 2003). The open-world allows players the freedom to move and act freely within the game environment, instead of following predefined paths and action sequences (Blizzard Entertainment Inc., 2012). While research has documented how such games can be used to engage and inspire students to take interest in science, technology, engineering, and mathematical (STEM) subjects (Steinkuehler & Duncan, 2008), the field is beginning to explore how they can be used for assessment. The extended capabilities provided in MMORPGs allow for a new, innovative approach to assessment. Unlike traditional assessments, which rely on students providing itemized feedback, assessment through MMORPGs can be captured in-situ, during game play. In this paper, we will describe how unobtrusive learning assessment techniques coupled with machine learning algorithms can be utilized to record a student's in-game actions and formulate a model of the student's knowledge without interrupting the student's game play. We introduce "Experiment Centered Assessment Design" (XCD), a framework for structuring a learning assessment feedback loop. XCD builds on the "Evidence Centered Assessment Design" (ECD) approach (Mislevy & Haertel, 2006), which uses tasks to elicit evidence about a student and his learning. XCD defines every task as an experiment in the scientific method, where an experiment maps a test of factors to observable outcomes. This XCD framework was applied to prototype quests in an educational MMORPG, *The Radix Endeavor*, being developed at The Education Arcade at the Massachusetts Institute of Technology. In the following sections, we provide background and context by first describing *The Radix Endeavor*. We then present an overview of learning assessment through Evidence Centered Design. Next, we describe the Experiment Centered Design assessment framework.

Then we provide examples of Experiment Centered Design used in *The Radix Endeavor* quests. Finally, we conclude with further ideas to expand Experiment Centered Design.

II. BACKGROUND & CONTEXT

The Radix Endeavor: An MMORPG for STEM

The Radix Endeavor is a sandbox MMORPG being developed at The Education Arcade at Massachusetts Institute of Technology. The game is set on a mysterious cluster of islands. The people of these islands live in a time reminiscent of the Middle Ages, when science and technology were limited. Furthermore, the government suppresses the population's ability to practice science in order to maintain control over them. Players are recruited to a secret society that finds scientific discoveries to undermine and overthrow the oppressive regime (Klopfer, 2011).

The learning goal of *The Radix Endeavor*, or *Radix* for short, is to engage high school students in learning mathematics and biology. Students assume different character roles that determine the curriculum of quests their character will need to complete. The structure of the open-world MMORPG offers players two important freedoms. First, a sandbox RPG gives players the freedom to explore the game world. Players have quests in various locations on the island, but players are not limited in where they go, what equipment to use, or what tasks to complete. Second, massively multiplayer online games foster open communication between players. Players are encouraged to share advice on solitary missions, and compelled to communicate with each other during multiplayer quests (Klopfer, 2011).

Radix also aims to use learning assessment to offer feedback to students while the game is played. MMORPGs have a multitude of information to convey about a player's status, abilities, location, equipment, enemies, and achievements. All of this information informs the player about his character and progress, and can be presented in a variety of interfaces. These interfaces allow a player to witness and analyze his position in the game and empowers players to make informed choices on the most effective path to success.

Even with efficient interfaces, however, a player may still become stuck. The ability to detect when a player is stuck allows the game to offer advice as needed or when it is most applicable. When a player has become stuck, the game should gradually assist the player in reaching his goal. By trusting and respecting a player's ability to progress with minimal help, the game preserves an environment that encourages exploration and learning (Gee, 2007).

The ability to offer analysis and feedback to students as they perform tasks can help the students self-diagnose their strengths and weaknesses. MMORPGs excel in actively engaging students through fantastical roles and scenarios. A distinction between game tasks and assessment tasks would break this illusion and disengage students. Thus, MMORPGs need what Shute & Ventura (2013) call *stealth assessment* — assessment that integrates evidence of learning competencies into game tasks. Hence, a student's work on a task in-game reveals evidence about their knowledge of particular subjects (Shute & Ventura, 2013).

Learning Assessment

The ability to record, summarize, display, and improve a player's progress requires learning assessment. Assessment measures a user's understanding of their situation, forms a model of

that student's knowledge, and acts on this model to further the user's goals. The techniques used to formulate this model and offer recommendations are an active area of research (Shute & Ventura, 2013).

The traditional form of learning assessment occurs in the classroom. A teacher teaches his students and measures their knowledge by administering tests. Teacher-made assessments rarely tell teachers what they need to know about their students' thinking (Black & Wiliam, 2009). Further, they often provide little in the way of feedback for students to regulate their learning. Formative assessments have the potential to provide important feedback to both teachers and students. This feedback is critical in helping teachers adapt instruction so students can overcome any misconceptions they have in moving along a learning progression. Yet, to do this, teachers need tools to adequately identify, measure, and evaluate what individual students know and do not know during the act of learning. Without the aid of technology, this may be difficult to accomplish during classroom instruction. Digital assessment can benefit teachers and students by offering tighter feedback loops that correlate a student's performances with their academic strengths and weaknesses (Shute & Ventura, 2013).

Evidence Centered Design

Evidence Centered Assessment Design (ECD) is an approach to constructing educational assessments that focus on measurable evidence of a student's learning. ECD collects and analyzes evidence from tasks performed by the student. Collectively, these student, evidence, and task models form the Conceptual Assessment Framework (CAF). Refer to Figure 1 below. The following paragraphs briefly describe each of the CAF models and how they could be applied to traditional classroom learning assessment (Mislevy, Almond, & Lukas, 2003).

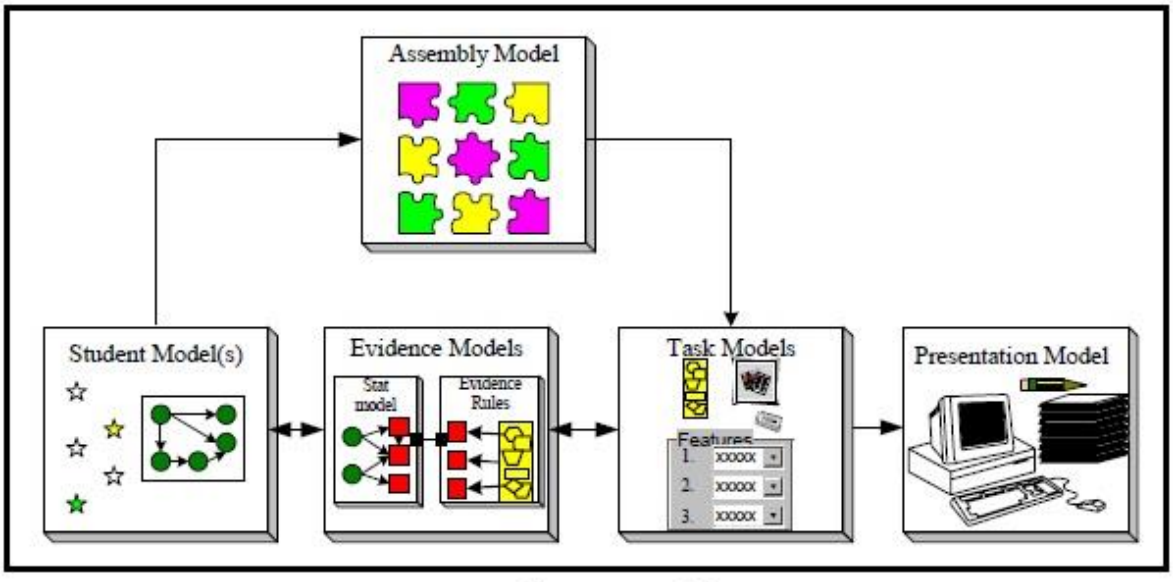

Figure 1: Distinct models of the Conceptual Assessment Framework

Student Model

First, the student model is a representation of a student's knowledge of a particular domain. For this example, the student model represents a student's understanding of biology. While it is impossible to gain an exact depiction of a student's knowledge, one must approximate and represent this knowledge. Hence, the student model is traditionally represented by a numeric grade (0-100) and simplified into the letters A, B, C, D, and F. In short, the student model asks *what competency are we measuring* (Mislevy, Almond, & Lukas, 2003)?

Evidence Model

Second, the evidence model provides instructions on how the student model should be updated given the result of a student's work on tasks. The evidence model has two parts. Evidence rules identify and summarize the meaningful work that shows evidence of learning. The measurement model accumulates and interprets this meaningful work to update the student model. In this example, the evidence model is the comparison of a student's test answers with the solutions. The rubric provides the evidence rules that label a student's work as correct or incorrect. The weight and impact of the exam on the student's grade (the student model) is the measurement model. In short, the evidence model asks *how do we measure competency* (Mislevy, Almond, & Lukas, 2003)?

Task Model

Next, the task model is composed of the scenarios that can elicit the evidence needed to update the student model. In this example, the tasks are the questions on a particular exam, which could be multiple-choice or open-response. The student's answers are the output of the task model and the input to the evidence model. In short, the task model asks *where do we measure one's competency* (Mislevy, Almond, & Lukas, 2003)?

Assembly Model

The assembly model structures the balance needed to gain an accurate student model from the family of tasks in the task model. In this example, a teacher must balance the question content and types before administering the test. The information obtained from an open response question may be more content rich than that of multiple-choice questions, and the teacher must determine what selection of questions is sufficient. In short, the assembly model asks *how much do we need to measure* (Mislevy, Almond, & Lukas, 2003)?

Presentation Model

The presentation model is defined by the medium that the tasks are delivered to the students. In this example, the exam may be administered with pencil and paper, through a computer interface, or even orally. The presentation model should not encumber the student and easily facilitate the assignment of tasks and collection of evidence. In short, the presentation model asks *how do the tasks look* (Mislevy, Almond, & Lukas, 2003)?

Delivery System Model

Finally, the collection of the student, evidence, task, assembly, and presentation models define the delivery system model. This model is intended to capture any issues not previously described by another model, such as the timing or security of the system. In this example, it is likely that the test is time-constrained and students are separated to avoid cheating. In short, the delivery model asks *how does the system work* (Mislevy, Almond, & Lukas, 2003)?

The Assessment Cycle

The ECD framework is intended to work within an assessment cycle defined by four key processes. This cycle defines the flow from selecting, displaying, performing, and scoring a task. Refer to Figure 2 below.

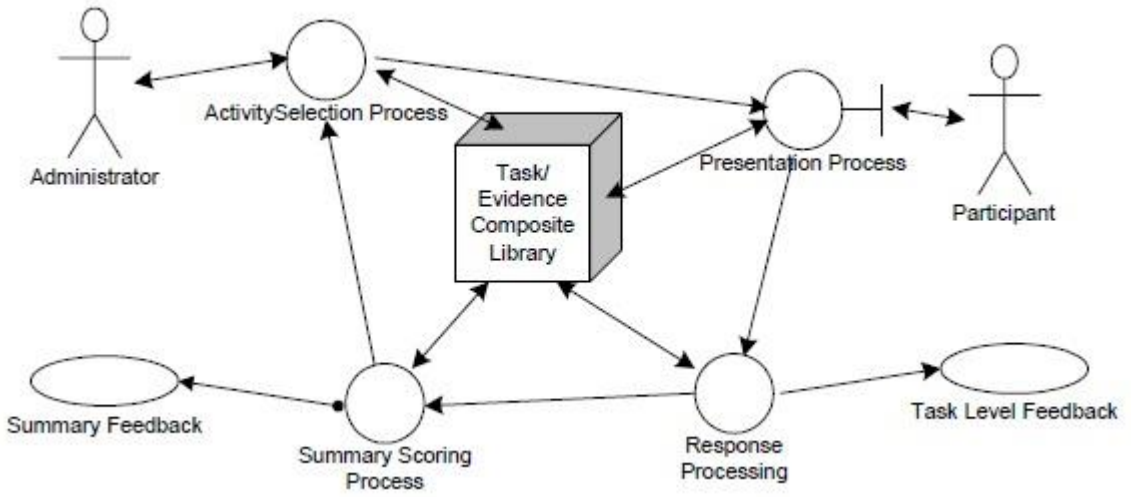

Figure 2: Assessment cycle processes and participants

The cycle begins with a set of tasks being selected from a large library of tasks by an administrator. Tasks are presented to the participant to work on. Upon completion, the participant submits his responses for processing. Processing a response includes interpreting a student's answer as well as any metadata captured from the student's work. Feedback about particular tasks can be reported to the student, teacher, or other interested parties. A response is also used to update the cumulative score of the user. Feedback that summarizes the student's overall score can be pulled from this summary scoring process. The scoring process updates the model of the participant's knowledge, and a new set of tasks is selected for the participant. The cycle may continue until a significant model is formed about the participant (Mislevy, Almond, & Lukas, 2003).

In a traditional classroom assessment, the teacher selects all of the questions for a test ahead of time. The test presents these questions to students, which captures their work and answers. The teacher scores the tests, updates the students' grades, and identifies which subjects to teach and review for the future.

Using ECD with digital technology can enhance the process of selecting tasks and aggregating results. For instance, imagine individually presenting questions to a student, where each new question depends on the student's answer to the previous question. This process could

identify the student's strong and weak subjects by dynamically avoiding topics that the student has mastered and focusing on questions that probe the student's weaknesses.

Back-end Assessment

Aggregating results can leverage many advantages of digital technology as well. Data mining is the process of analyzing large amounts of logged data for trends and patterns. By applying data mining processes to evidence collected in ECD, students' statistics and patterns can easily be brought to educators' attentions. A variety of back-end assessments, including item response models and artificial neural nets, can offer various levels of insight and interpretation. Various researchers have been exploring how powerful these algorithms can be for assessments with digital technologies (Shute, Masduki, Donmez, 2010; Shute, 2011; Quellmalz, Timms, Buckley, Davenport, Loveland, & Silberglitt, 2011; Behrens, Mislevy, Dicerbo, & Levy, 2012; Williamson Shaffer & Gee, 2012; Sao Pedro, Baker, Gobert, Montalvo, & Nakama, 2013; Baker & Clarke-Midura, 2013; Clarke-Midura & Yudelson, 2013).

With the exception of Sao Pedro et al, all of these projects have started with ECD, and modified it during the design process to fit their particular needs. As we started out designing our back-end assessment for *Radix*, we realized that the open nature of the world centered on complex experiments. Previous frameworks failed to compensate for the freedom of choices available to players of an open-world game as well as the cumulative value of these choices. The need to stealthily capture multiple, interconnected actions between the player and the game world propelled us to modify ECD to fit the demands of an MMORPG.

III. DESIGN

The Radix Endeavor aims to use in-game assessment and machine learning to capture and display students' knowledge for various audiences including teachers, students, and researchers. A number of quests in *Radix* have already been prototyped with a variety of variables and contexts. In order to use machine learning techniques, a standard task model must be structured to accept input from players and provide output to the evidence model. Experiment Centered Design (XCD) is a modification of Evidence Centered Design that templates quests developed for *The Radix Endeavor*. Recording quests and applying machine learning techniques requires deconstructing every quest into a shared, standard format.

Structure of a Quest

Defining the common quest elements is the first step to finding a uniform quest structure. Every quest must have triggers and mechanisms that allow the player to start, work on, and complete tasks. Figure 3 shows these stages of a quest. In each stage, information is flowing between the player and the game, or the player is analyzing the information it has gathered.

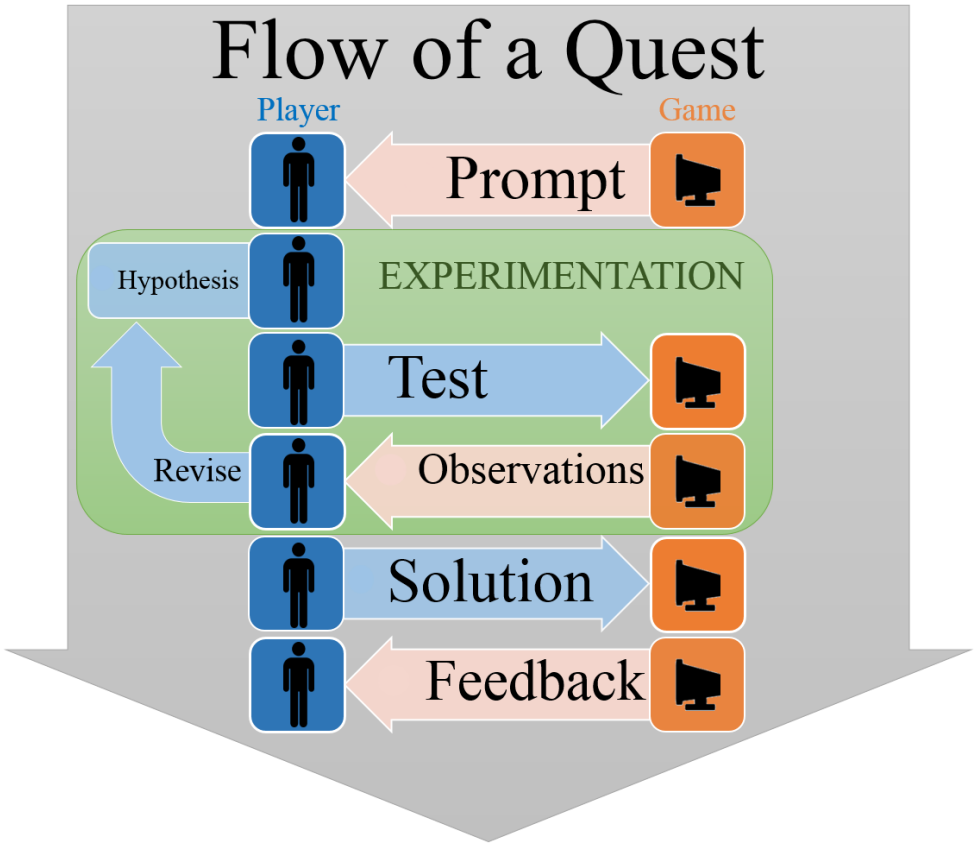

Figure 3: Flow of information in a quest

Prompt

As we see in Figure 3, every quest begins with a prompt. The primary purpose of the prompt is for the game to inform a player what the victory condition of the quest is. The prompt often contains instructions on where and how to approach the quest. This dialogue can be delivered through various sources, including non-playable characters, magical objects, or simple popup directions.

Experimentation

Second, every quest requires experimentation. The goal of quests in *The Radix Endeavor* is to encourage scientific inquiry. Part of achieving this goal is showing that experimentation is a useful skill that is applicable to a variety of situations. In the world of *Radix*, players utilize specific skills for specific experiments, but learn the pattern of conducting steps of the scientific method: form a hypothesis, conduct a test, and analyze results.

A hypothesis is an assumption that attempts to explain a particular phenomenon. A hypothesis is formed from one's knowledge of the domain, which may be empty or expansive. This knowledge is essentially the student model, an unknowable state of comprehension teachers wish to understand.

A test is an experiment done to support or refute the hypothesis. The set of variables used in the experiment are called factors. These factors are acted upon by an operator. An experiment

results in a set of observations visible to the user. In short, an experiment is action taken on a set of factors that produces observations.

The observations produced from an experiment may support, contradict, or offer no information about the hypothesis. Analyzing these observations requires separating which observations are conclusive. These conclusive observations build on the knowledge of the domain and allow one to affirm or adjust one's hypothesis. These observations may also be used as factors in future experiments. The cycle of hypothesizing, testing, and analyzing continues until the experimenter is confident that their hypothesis is the truth.

Educational games that allow users to conduct experiments vary in how they present the steps of the scientific method to the player. Many presentations explicitly reveal these steps. For example, some games require players to input text describing the reasoning behind their hypothesis before they can conduct the experiment. Other games associate a particular location as a kind of "headquarters" for conducting experiments. Still other games provide players with charts of the relevant observations after an experiment. All of these options lessen the immersive experience by bringing the scientific method to the forefront of the player's attention. *The Radix Endeavor* supports an immersive experience by allowing users to conduct experiments anywhere in the world unimpeded by questionnaires and read-outs. The quest prototypes provided for *Radix* are built around tools that players can carry in their inventory to any part of the game world. Each of these tools has different operations, which include measuring, probing, and creating objects in the environment. In short, these tools allow users to conduct experiments according to the scientific method. *Radix* attempts to use a player's actions and experiments to build the student model for that player.

Solution

Third, every quest has one or more solutions. A solution is a condition that marks the completion of a quest. Solutions may be a set of predetermined values. A simple example is the binary set "true" or "false." Solutions can also be open responses with limited constraints, such as choosing a number within a range. It is important to note that a quest may have multiple solutions, and solutions may be correct or incorrect. Players attempt to complete a quest by submitting responses that match a solution. Responses that trigger a solution are valid responses. Responses that do not match a solution and therefore do not complete the quest are considered invalid responses. These valid and invalid responses offer insight into a student's understanding.

As an overly simplified example, consider the question "What is the sum of adding one plus one?" This question can be solved by entering any numerical value, but the correct response is the number "2". Any other numeral, like "11," is an incorrect response. An invalid response is the word "two," because it is not a numeral and cannot be accepted. Solutions in *Radix* are more complex and can require the manipulation of the environmental or the fulfillment of multiple conditions in order to submit a response. Yet every response is either correct, incorrect, or invalid.

A majority of quests are intended to teach students particular educational subjects. A learning concept is knowledge or behavior being taught or exercised in such a quest. These concepts may be correct, unusual, misguided, or simply interesting patterns of thinking or acting that are observed by the game while the student plays. A quest may have more than one learning concept associated with it. A valid response that matches a correct solution to a quest implies that a student understands the educational content associated with that quest. A learning

objective is knowledge or behavior that is correct. In the example above, the learning concepts focus on addition. The correct response, "2," suggests that a student has met the learning objective of understanding addition.

Quests are designed for students to ultimately succeed. However, students are expected to err while experimenting. A misconception is an error in judgment about an educational subject. Incorrect or invalid responses may reveal misconceptions that the student has about the quest and its learning concepts. In the example above, the incorrect response "11" reveals a misconception about the addition operator. The invalid response "two" reveals a misconception about non-numerical responses being acceptable.

Feedback

Finally, every quest must provide feedback for responses. This feedback has two primary audiences: players and educators. The different audiences require different feedback. Players desire information that advances their progress in the game. In order to maintain an immersive environment, this information must also be provided in thematically appropriate contexts. These requirements imply that players need quest feedback to be given after reaching learning objectives or falling into misconceptions. The feedback given to students may reveal these concepts in an explicit summary, or may offer gradual clues on how to proceed.

Educators desire information that summarizes the learning objectives and misconceptions uncovered by students. Teachers use this information to guide students in-game and in the classroom. Educational researchers appreciate the ability to study particular patterns among students' quest habits. For these reasons, educators require visible, easy to interpret feedback that summarizes students' submissions after they attempt quests. In short, students need immediate feedback while experimenting in a quest, while educators need a timeline of student actions that they can review quickly.

In summary, the core elements of a *Radix* quest are prompts, experiments, solutions, and feedback. All of these elements pass information between the game system and the user. However, prompts and feedback send information in one direction from the system to the user. Experiments and their solutions offer a dialogue between the player and the system. Assessing this dialogue offers insight into the knowledge, intent, and patterns of the player.

Object Models

One way to represent the structure of a quest is through an object model. An object model is a diagram that defines a mapping of entities and their relationships between one another. Refer to Figure 4. In the figure, every phrase surrounded by a box represents a set. An underlined phrase means the set is an abstract set. Every arrow represents a relationship between sets. Arrows with open heads define a "subset" relationship. Arrows that share an open head separate two or more disjoint subsets. Arrows with filled heads define multiplicity relationships. The direction and name of the relationship defines the relation between the sets. For example, an arrow from set S to set T with the name "owns" specifies that a set S owns a set T and that a set T is owned by a set S. Multiplicities specify how many sets map to another set. The multiplicity symbols represent at most none (*), at most one (?), at least one (+), and exactly one (!). If no symbol is specified, the relationship is implied to mean at most none (*). An "attribute" relationship maps at most none (*) of set S to exactly one (!) of set T (Jackson, 2012).

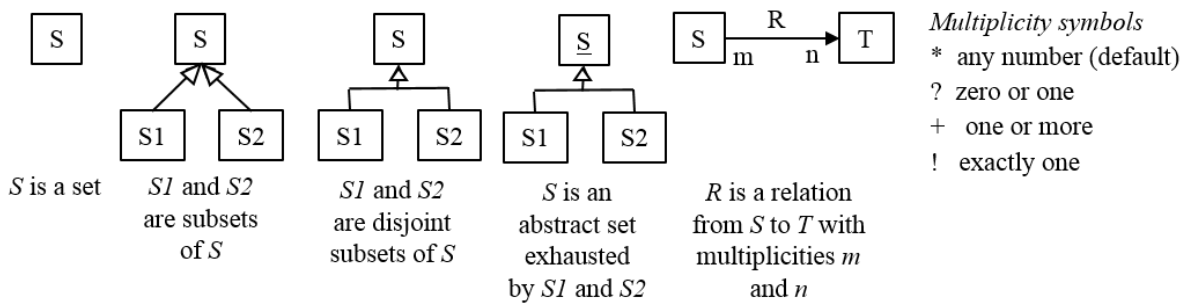

Figure 4: Representation of sets and relationships in an object model

An object model defines constraints between sets and relationships. An individual "instance" of an object model is a structure that follows the object model template with specific entities. There are an infinite number of object model instances that could follow the template of a single object model.

Databases are essential for recording and storing the multitude of information in digital games, including information about players, quests, and the environment. Object models are useful for structuring database schemas. Database tables store objects and relationships that map objects one-to-one, one-to-many, and many-to-many. The sets and multiplicities of relationships in an object model define the tables of a relational database. Instances of the object model fill entries in these database tables.

As a simple example, consider the object model in Figure 5 that captures the relationships between faculty, students, and classes. (Jackson, 2012) The student body is partitioned into visiting students and regular students, where regular student must enroll in exactly one degree program. All students must register in at least one class and have exactly one advisor. Faculty includes two subsets: advisors and teachers. Advisors advise any number of students, and teachers teach at most one class. Finally, every class must have at least one teacher.

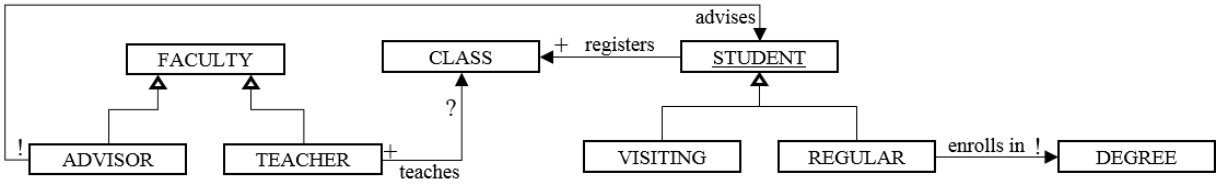

Figure 5: College registration object model

Figure 6 is one instance of the object model provided in Figure 5. Alice is a regular student registered in calculus and enrolled for a mathematics degree. Bob is a visiting student registered in biology. Mr. Beta is an advisor for Alice. Mr. Gamma and Mr. Delta are teachers of calculus. Mr. Epsilon teaches biology and advises Ben. Finally, Mr. Alpha is a faculty member who is neither a teacher nor an advisor. All of the constraints of the object model from Figure 5 are followed.

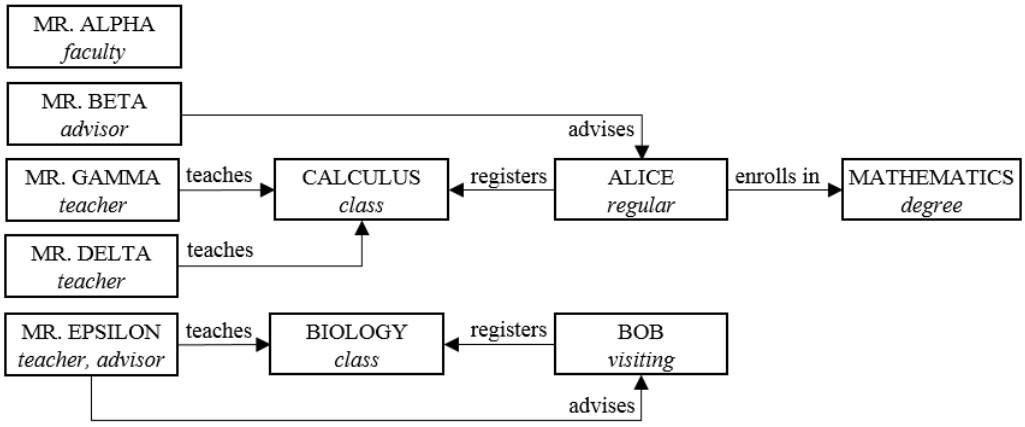

Figure 6: College registration object model instance

Furthermore, the data captured in Figure 6 completes the database in Figure 7. This database allows for the easy querying, creation, deletion, and revision of new and existing students, faculty, and classes.

FACULTY	
ID	NAME
1	MR. ALPHA
2	MR. BETA
3	MR. GAMMA
4	MR. DELTA
5	MR. EPSILON

STUDENTS	
ID	NAME
1	ALICE
2	BOB

CLASSES	
ID	NAME
1	CALCULUS
2	BIOLOGY

DEGREES	
ID	DEPARTMENT
1	MATHEMATICS

ADVISES		
ID	TEACHER ID	STUDENT ID
1	2	1
2	5	2

TEACHES		
ID	TEACHER ID	CLASS ID
1	2	1
2	3	1
3	5	2

ENROLLMENTS		
ID	STUDENT ID	DEGREE ID
1	1	1

REGISTRATIONS		
ID	STUDENT ID	CLASS ID
1	1	1
2	2	2

Figure 7: College registration database tables

Quest Template Object Model

By abstracting the structure of quests, the quest elements and their relationships can be captured in an object model. Figure 8 is an object model that connects the users, quests, experiments, and educational content. The sets included in Figure 8 depict elements required by all quests in *The Radix Endeavor* (Clarke-Midura, 2012).

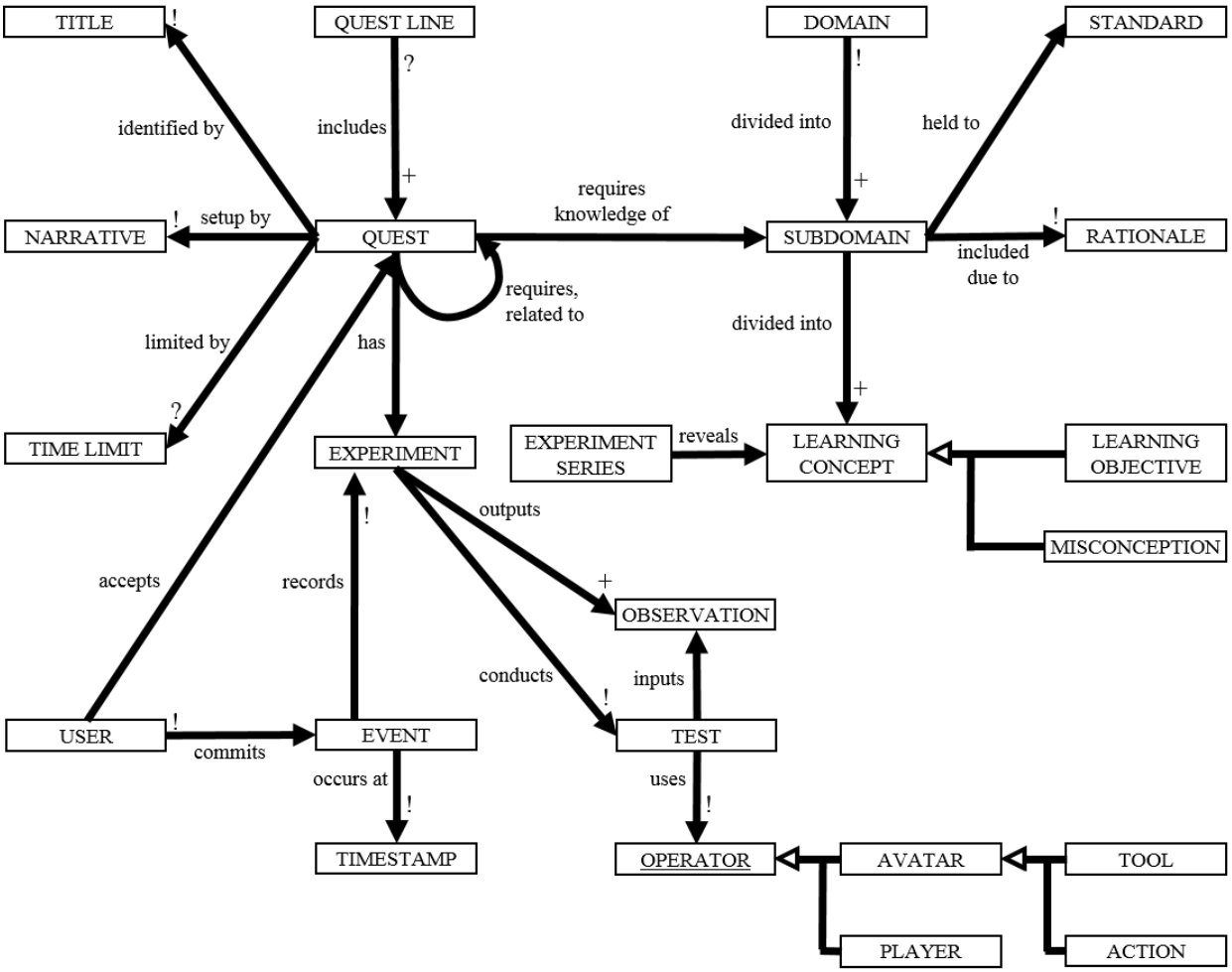

Figure 8: Quest template object model

Every quest is identified by its title, as well as a narrative that establishes the fiction of the quest. A quest may belong to at most one quest line, and quest lines must include at least one quest. It is possible that quests may require a time limit, or other attributes that are not shown in Figure 8. Finally, quests may require or be related to other quests.

A domain of knowledge is divided into subdomains, and every subdomain belongs to only one domain. Completion of a quest may require knowledge of any number of subdomains. Every subdomain is included in *Radix* because of some rationale. These subdomains are likely held to particular educational standards. Finally, every subdomain has one or more learning concepts attached to it. These learning concepts may be learning objectives or misconceptions, as described previously.

Quests may involve conducting any number of experiments. An experiment is a particular mapping of a test to at least one observation. As described earlier, a test is an operation on a number of factors. This operation can be performed by either the real-world player or his in-game avatar. For instance, solving a mathematics problem could be done with in-game tools or by the player's calculator in the real world. The more operations that a player executes in-game, the more data can be captured about the player's experimentation. The operations an avatar can perform include using tools or executing actions. A test may input and operate on any number of observation factors. However, the experiment always outputs at least one observation.

Users accept quests. When a user conducts an experiment, the experiment is logged as an event, which is marked with a timestamp. These event logs can be easily retrieved and filtered to study the activity of the user. Users may have other attributes, such as their name and level, which are not currently represented in this object model.

An experiment series is a particular pattern of one or more experiments. Experiments, when performed individually or in a specific order, may reveal certain behaviors. Finding these patterns in a player's event history implies that the student exhibits that behavior. Filtering through database queries offers one solution, which requires optimizing the database to handle these search queries.

While Figure 8 gives a broad overview of important quest elements, particular sets are more interesting to different people. For example, the game writer may be interested in the quest and user attributes, but not the educational content. Curriculum developers may be solely interested in the educational content, but not how the quests, experiments, or users are involved. This paper concerns itself with learning assessment. Therefore, Figure 9 is a simplified object model that keeps the sets vital to assessing a user's educational progress.

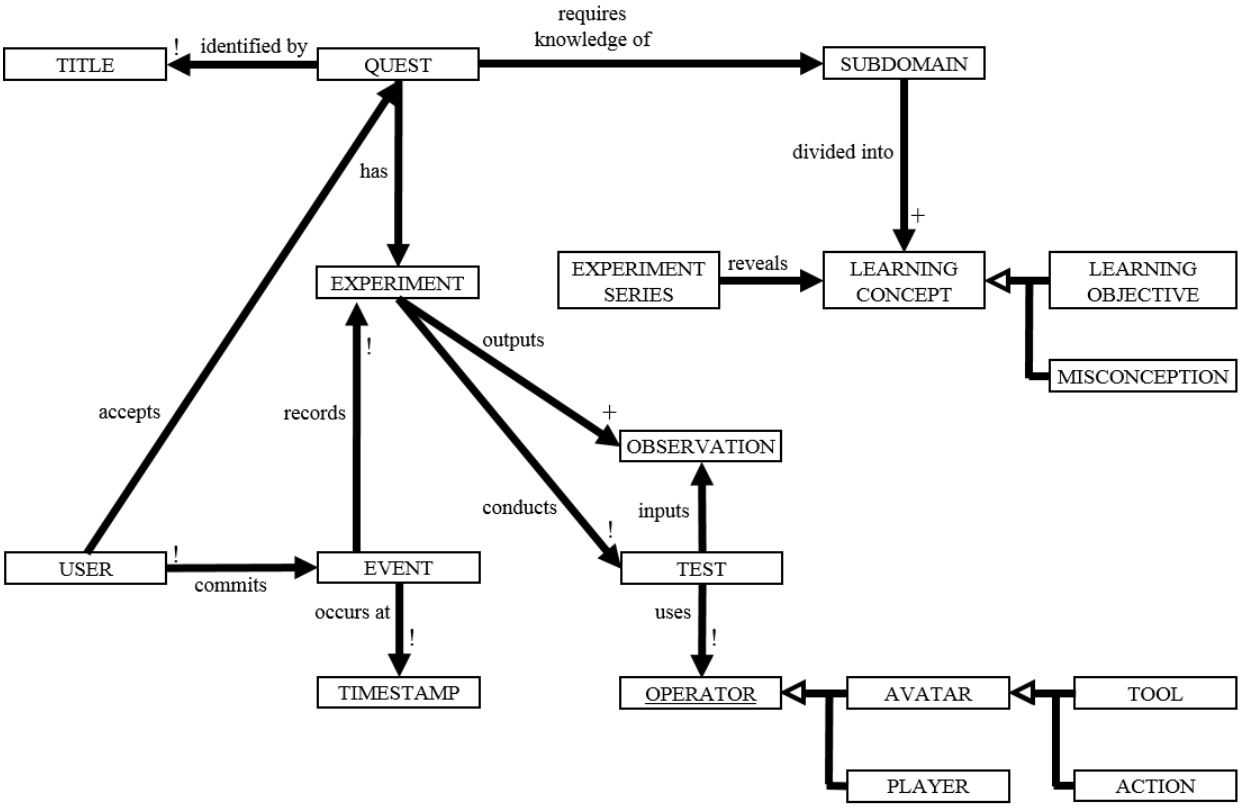

Figure 9: XCD object model

The central set that ties this object model together is the experiment. Quests are designed around recreating experiments. Users perform experiments. Learning concepts are revealed by series of experiments. Experiments are key to designing, playing, and learning from quests.

Features of Experiment Centered Design

Experiment Centered Design, like Evidence Centered Design, is used as a framework to guide development. XCD is intended to be used during the development of quests and quest lines. In this way, Experiment Centered Design has a few unique properties.

First and foremost, XCD can replicate the ECD system by treating open- and closed-response questions as types of experiments. Players are prompted to answer a question, and the experiment is the student's ability to select the correct answer from the set of all possible answers. In this way, XCD is capable of administering questionnaires and emulating ECD, although this method is discouraged given the opportunity to use stealth assessment.

Second, XCD allows different designers multiple affordances to prototype experiments for educational study. Game designers may start with operations they want in-game players to perform. Curriculum designers may start with what experimental observations would reveal learning concepts. Different designers can iterate separately or work together to formulate engaging and informative experiments.

Third, a series of experiments can uncover knowledge about a player's thought process that individual responses fail to capture. Players are encouraged to perform multiple experiments in order to learn about the world and solve complex problems. The experiments and the order in which they are performed might follow patterns that reveal misconceptions or learning objectives.

Finally, experiments offer variability unexplored in traditional task models. The world of *Radix* is a complex system of objects, environments, and characters that players are encouraged to interact with. Players can experiment in the world and input consistent factors, but randomness in the system causes variable outputs. Analyzing how a student adapts to these outcomes with more experimentation offers rich insights into their strategy and learning.

IV. IMPLEMENTATION of XCD

By applying the benefits of Experiment Centered Design to *The Radix Endeavor*, researchers were able to capture learning concepts in quests and experiments that were developed by independent game designers. The following examples show the object model instances and learning assessment built around experiments.

Volume and Surface Area

A quest line was developed for students to complete cost optimization problems to learn about volume and surface area. In game, players are provided with a tool to construct rectangular prisms. A non-playable character, an animal breeder, needs to travel with his animals in boxes. The first quest asks players to create a box that can fit a "tropical chicken" ten units wide, twelve units long, and fifteen units tall. If the player submits a box that is too small, the player must try again. If the player submits a box that is greater than or equal to the specified dimensions, the box is accepted.

This quest offers an interesting look into how different developers can utilize Experiment Centered Design to assess a player's progress. Assume there are two developers: a game designer who is interested in players' general game progress and a researcher who is interested in closely capturing student actions. Figure 10 shows a player creating and submitting a box to the

animal breeder. In this interaction, the researcher focuses on the act of creating boxes as the experiment, which a player does in the first two panels. The game designer focuses on the act of submitting a box as the experiment, which the player does in the last two panels.

Figure 10: Player completing "Chicken Box" quest.

Assume there are two players: Alice and Bob. Alice creates a box that is 10 x 12 x 15 units. She submits the box and the breeder is happy to accept it. Bob creates a box that is 1 x 12 x 15 units. He submits the box, but the breeder rejects it. Bob creates another box that is 11 x 22 x 16 units and the breeder is happy to accept it.

Game Designer Object Model

The game designer treats the act of submitting a box as the experiment. He discretizes the set of all possible submissions into three experiments: submitting a box that is optimal, too small, or too big. When a box has dimensions that match 10 x 12 x 15 units, the box is an exact match and the vendor accepting the box provides feedback to the player that he has given a correct solution. Performing this experiment also implies that a student has met a learning objective about

understanding volume and surface area. If any of the dimensions of the box are too small, the vendor rejects the box. Performing this experiment implies that a student has misconceptions about volume and surface area. Finally, if the box is bigger than the size of the chicken, the vendor also accepts it and the submission indicates a learning objective has been met, albeit a suboptimal solution. In essence, the game designer treats the quest like a multiple choice problem. Refer to Figure 11 below.

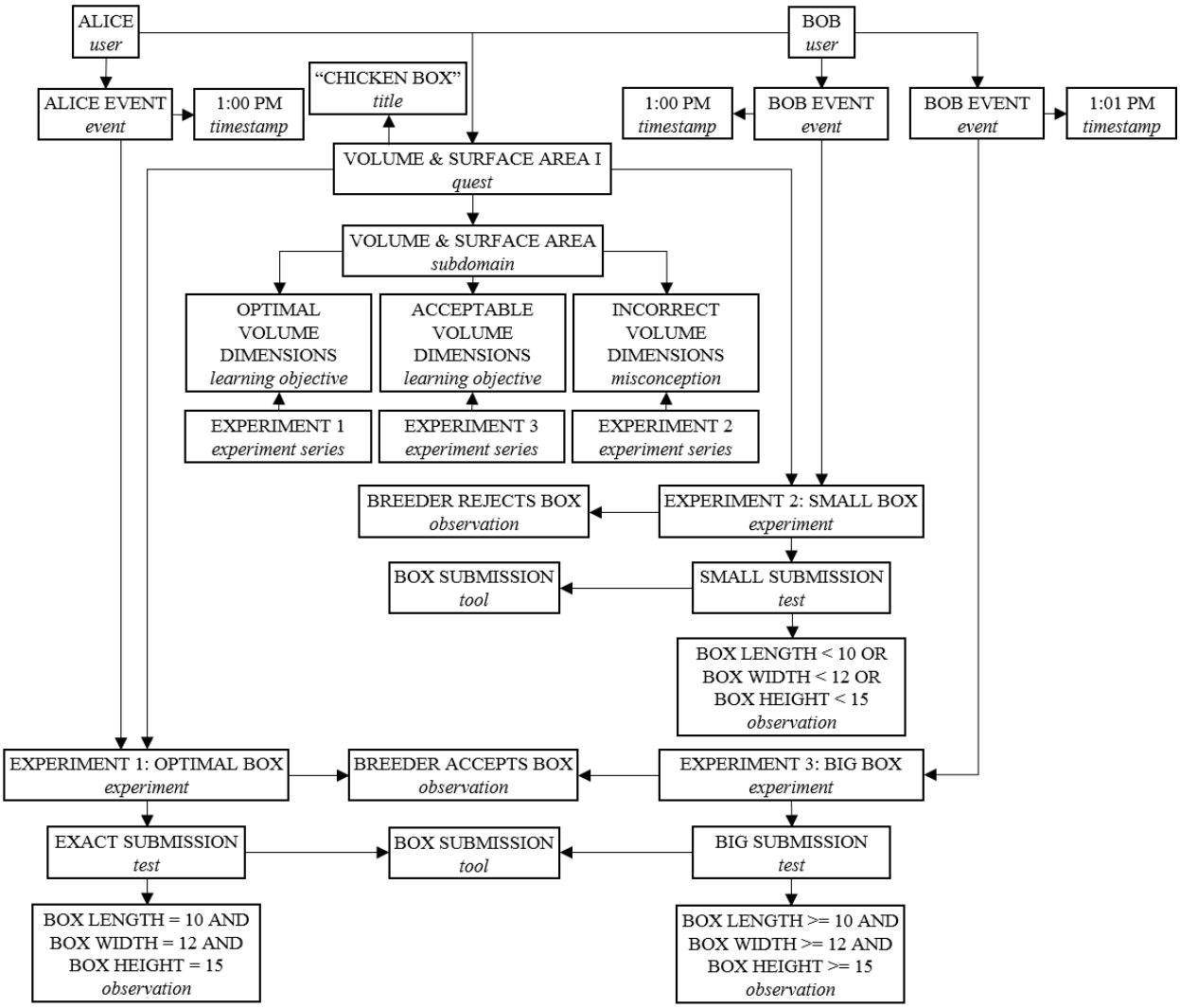

Figure 11: Game designer XCD focuses on player submissions

In the above instance, Alice performs experiment one, and the system recognizes that she submits an optimal solution. Bob performs experiment two, and the system flags him as having a misconception. Bob then performs experiment three, and the system records that Bob submitted an acceptable response. The system is knowledgeable of the number of submissions that Alice and Bob required to pass the quest and can differentiate optimal and suboptimal answers.

Researcher Object Model

The researcher is interested in the same learning objectives. However, the researcher wants to capture more specific information about the boxes created by players. To do so, he wants to record the dimensions of every box ever created. Every box maps to one of the learning objectives. In essence, the researcher treats the quest like an open-response problem. Refer to Figure 12 below.

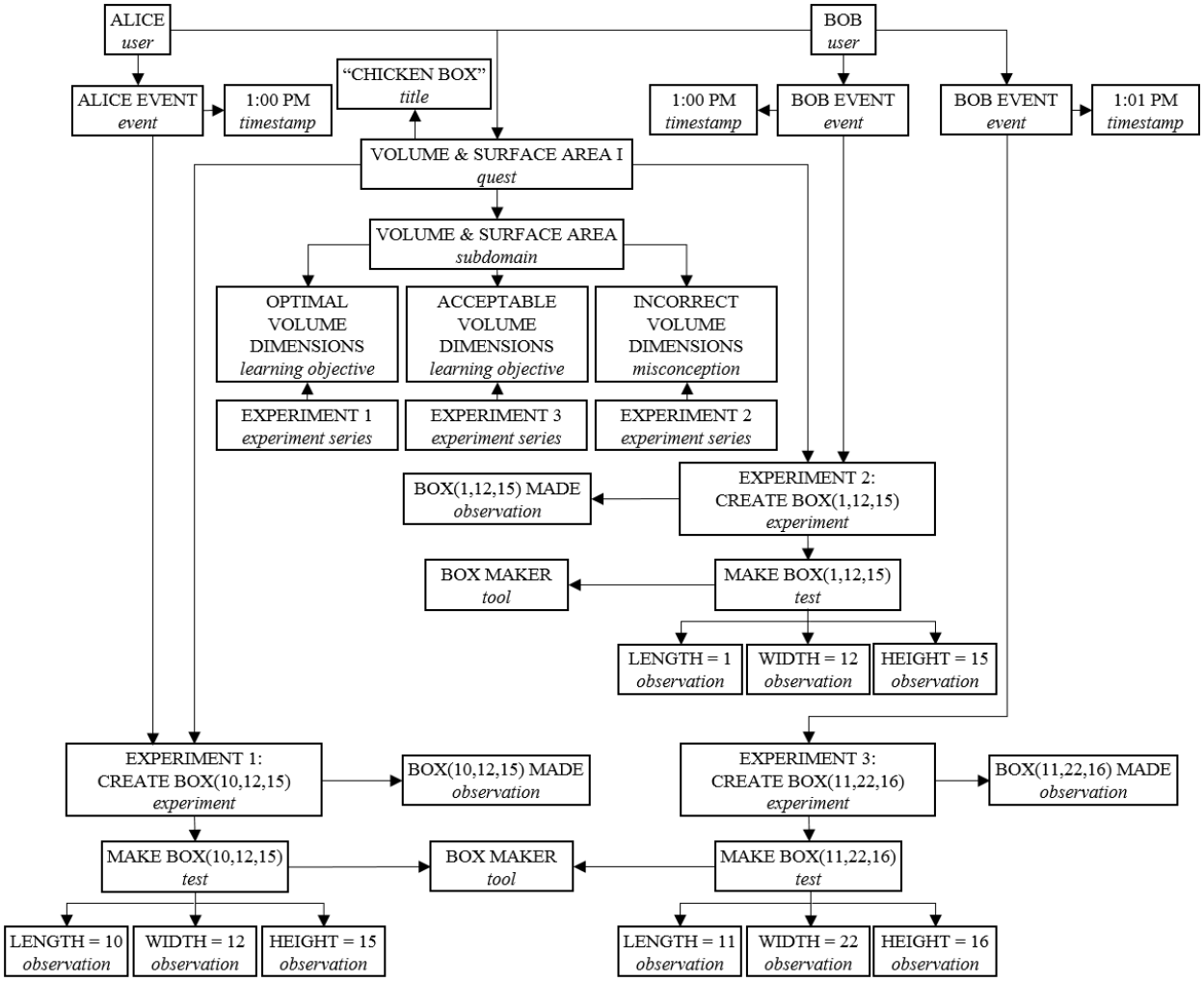

Figure 12: Researcher XCD focuses on player creations

The quest attributes and learning objectives are identical in this object model. However, the information about box dimensions is valuable to the researcher. The box maker tool allows students to input a length, width, and height value and outputs a box with those dimensions. In the above instance, Alice performs the first experiment. The system recognizes that Alice has created an optimal box. Bob first creates a box that is incapable of fitting the chicken. The system analyzes Bob's creation, recognizes it as being too small, and records it as a misconception. Bob performs another where the box is bigger than the chicken. The system evaluates this creation and records it as an acceptable box. The system is trained with Alice and

Bob's submissions. By associating more experiments with learning concepts, the researcher can tag other interesting phenomena, such as irregular or dyslexic submissions.

Merging Object Models into Database Storage

As previously stated, one of the benefits of XCD is that designers can work independently to iterate and evaluate quests and store this data in the same database schema. In the example above, the system can be loaded with both sets of experiments. When creating an experiment series to flag learning concepts, the game designer and researcher filter by the experiments that meet their particular needs. Refer to Figure 13.

QUESTS		
ID	NAME	TITLE
1	Volume & Surface Area I	"Chicken Box"

SUBDOMAINS	
ID	NAME
1	Volume & Surface Area

QUEST SUBDOMAINS		
ID	QUEST ID	SUBDOMAIN ID
1	1	1

LEARNING CONCEPTS		
ID	NAME	TYPE
1	Optimal Volume Dimensions	Learning Objective
2	Acceptable Volume Dimensions	Learning Objective
3	Incorrect Volume Dimensions	Misconception

SUBDOMAIN LEARNING CONCEPTS		
ID	SUBDOMAIN ID	LEARNING CONCEPT ID
1	1	1
2	1	2
3	1	3

EXPERIMENT SERIES	
ID	EXPERIMENT ID SERIES
1	1 or 4
2	2 or 5
3	3 or 6

LEARNING CONCEPT EXPERIMENT SERIES		
ID	LEARNING CONCEPT ID	EXPERIMENT SERIES ID
1	1	1
2	2	3
3	3	2

EXPERIMENTS		
ID	NAME	TEST ID
1	Optimal Box	1
2	Small Box	2
3	Big Box	3
4	Create Box(10,12,15)	4
5	Create Box(1,12,15)	5
6	Create Box(11,22,16)	6

TESTS		
ID	NAME	OPERATOR ID
1	Exact Submission	1
2	Small Submission	1
3	Big Submission	1
4	Make Box(10,12,15)	2
5	Make Box(1,12,15)	2
6	Make Box(11,22,16)	2

OPERATORS		
ID	NAME	TYPE
1	Box Submission	Tool
2	Box Maker	Tool

OBSERVATIONS	
ID	DESCRIPTION
1	Breeder Accepts Box
2	Breeder Rejects Box
3	Box Length = 10 & Box Width = 12 & Box Height = 15
4	Box Length < 10 & Box Width < 12 & Box Height < 15
5	Box Length >= 10 & Box Width >= 12 & Box Height >= 15
6	Length = 10
7	Width = 12
8	Height = 15
9	Box(10,12,15) Made
10	Length = 1
11	Box(1,12,15) Made
12	Length = 11
13	Width = 22
14	Height = 16
15	Box(11,22,16) Made

TEST OBSERVATIONS		
ID	TEST ID	OBSERVATION ID
1	1	3
2	2	4
3	3	5
4	4	6
5	4	7
6	4	8
7	5	10
8	5	7
9	5	8
10	6	12
11	6	13
12	6	14

EXPERIMENT OBSERVATIONS		
ID	EXPERIMENT ID	OBSERVATION ID
1	1	1
2	2	2
3	3	1
4	4	1
5	5	2
6	6	1

QUEST EXPERIMENTS		
ID	QUEST ID	EXPERIMENT ID
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	1	6

USERS	
ID	NAME
1	Alice
2	Bob

EVENTS			
ID	USER ID	EXPERIMENT ID	TIMESTAMP
1	1	1	1:00 PM
2	2	2	1:00 PM
3	1	4	1:00 PM
4	2	5	1:00 PM
5	2	3	1:01 PM
6	2	6	1:01 PM

Figure 13: Database tables that combine the object model instances of the game designer and researcher for the volume and surface area quest

Figure 13 stores the information from the object model instances provided in Figure 11 and 12. Some of the information in these figures is redundant, such as quest title and user names, and the database only needs to store this information once. The database stores the pertinent information about the creation and submission of boxes. When querying the database, the game designer filters entities that relate to the "box submission" operator while the researcher filters entities that relate to the "box creation" operator. Because each of these designers followed the XCD

framework, all of their information is captured and stored in a routine way that can be easily retrieved and manipulated.

Mendelian Inheritance

A quest was developed for students to recreate Gregory Mendel's breeding experiments in order to learn genetics. In the game, players come across a field of flowers with opposite traits A and B. The observable trait may be color, smell, or size, but the trait differs between players to prohibit students from giving the solution to other players. A local merchant agrees to pay the player if they can deliver the flower with the dominant trait. The player has no tool to determine the genotypes of these flowers, but he does have a tool that allows his avatar to cross breed two flowers and grow their children in a garden. When two flowers are crossed, however, they are destroyed in the process.

Experiment Object Models

The system has knowledge of the flowers' genotypes and knows that there are 16 possible experiments. Figure 14 is the object model instance for the experiments of this quest. The title of the quest is "Flower Breeding," and the biology subdomain it covers is genetics. In this instance, assume flowers with trait A carry the dominant gene. A garden with all "Type A" flowers has plants with identical phenotypes but either homogeneous dominant XX or heterogeneous Xx genotypes. Hence, a player viewing a flower of type A can view the phenotype of trait A but cannot be sure of its genotype XX or Xx. Assuming trait A is dominant, "Type Ab" flowers are always heterogeneous Xx, and "Type B" flowers are always homogeneous recessive xx. Every test uses the "cross" tool, which takes two parent flowers as inputs, crosses them, and produces a garden of children. The inputs are the flower types A, B, or Ab. The outputs of an experiment are the gardens of children, which contain either all A, all B, all Ab, mixed A and B, or mixed Ab and B flower types.

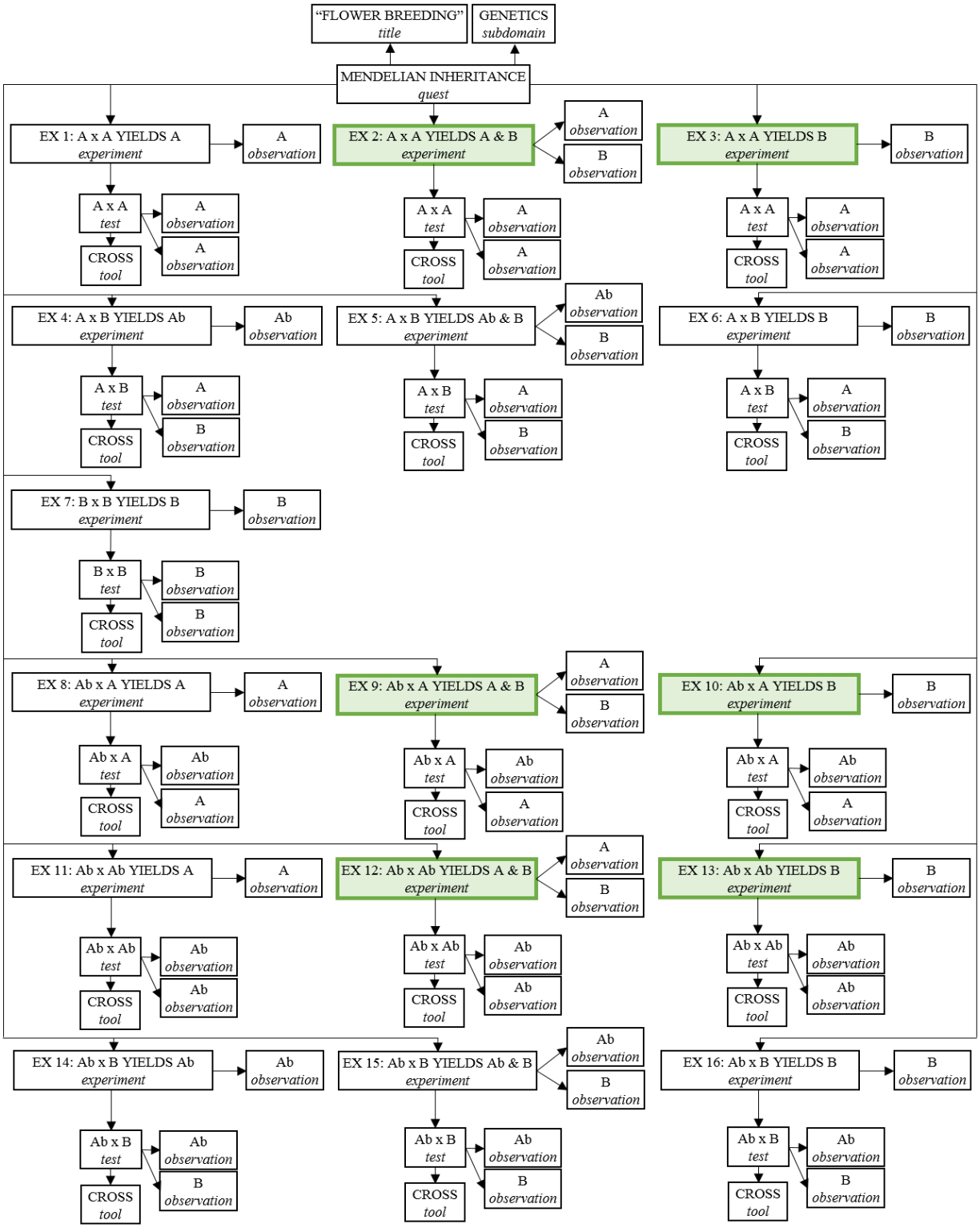

Figure 14: 16 Experiments of type A, Ab, and B flower breeding

Note that different combinations of parents can yield the same garden of children, but not every garden of children can be created from crossing two flowers. Refer to Table 1 below.

Table 1: Valid combinations of flower parents and offspring

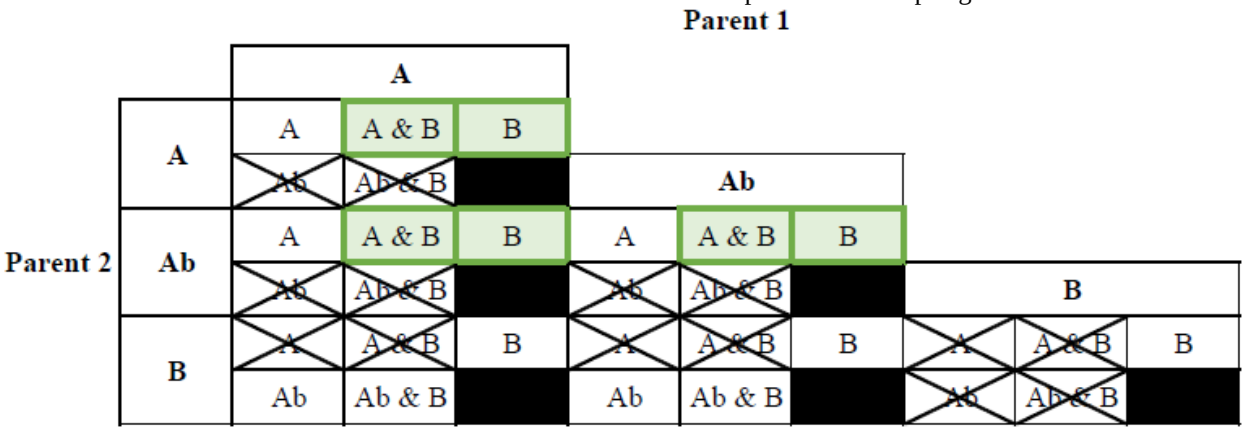

Table 1 shows the thirty theoretical combinations of flower parents and offspring. However, every cell of Table 1 that is crossed out cannot be bred. Some combinations are forbidden by the laws of genetics. Other combinations cannot be logically deduced. For example, when crossing two type A parents, one can never be certain that any of the children are type Ab, because crossing two type A parents means any of the following genotype crosses could have occurred: XX with XX, XX with Xx, or Xx with Xx. These crosses could yield children with any genotype.

Both Figure 14 and Table 1 highlight particular sets. Each of these sets represent an experiment when two parents with trait A are crossed and a child of trait B is produced. This phenomenon is the only proof that trait A is dominant. A player must perform one of these experiments as well as recognize this phenomenon in order to correctly complete the quest.

While Figure 14 is a verbose representation of each experiment, it contains multiple, redundant references to information. Figure 15 eliminates the redundant sets in Figure 14 while still mapping every relationship in the quest. This representation more accurately reflects how the database can compactly store a multitude of possible experiments that are performed by players.

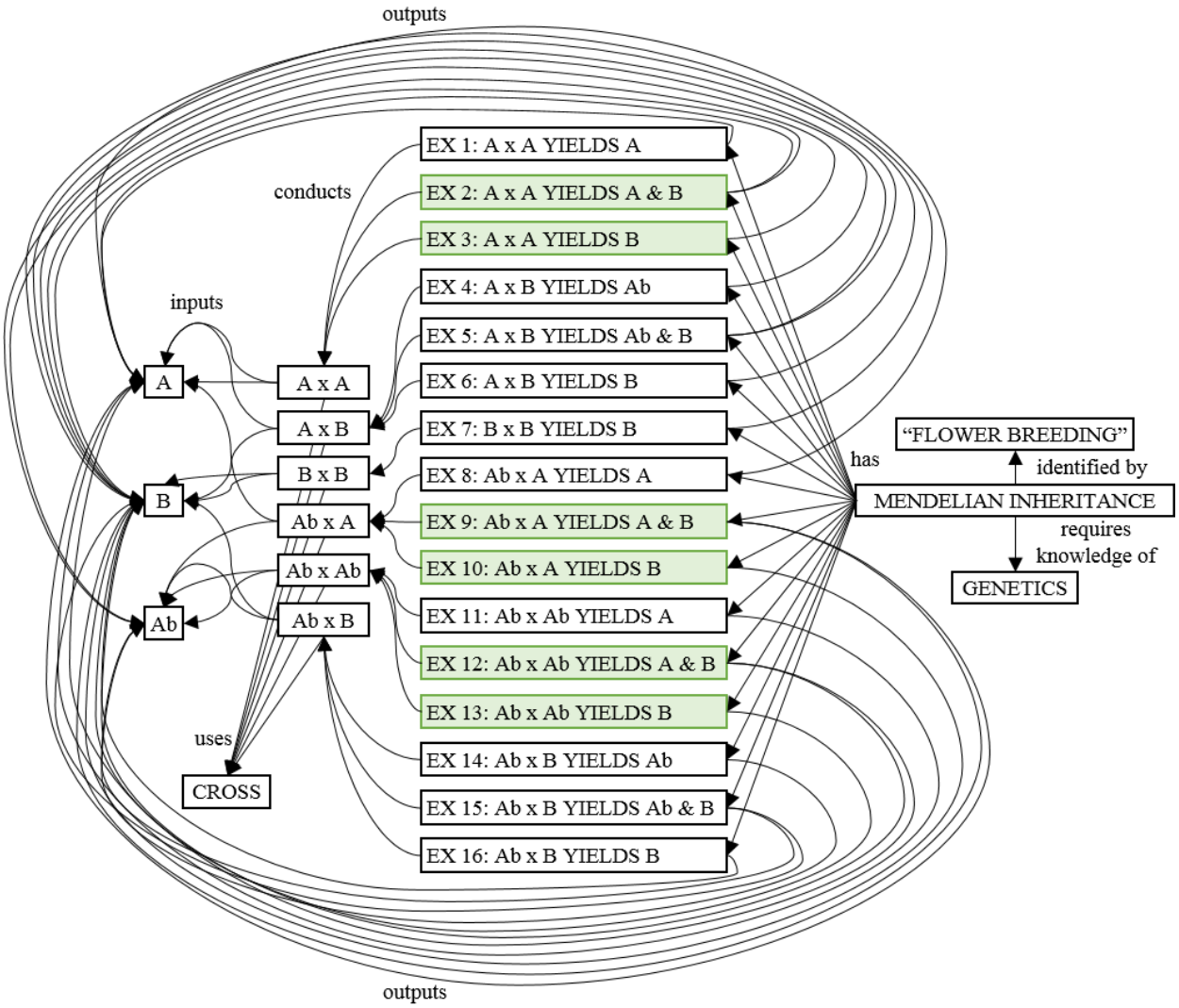

Figure 15: Concise view of type A, Ab, and B flower breeding

Flower Breeding Experiment Series

Due to the random nature of breeding, students are expected to perform multiple experiments. However, players could perform experiments in certain patterns that reveal particular learning objectives or misconceptions. For example, one series of experiments has a high probability of yielding the solution and indicates careful premeditation of the problem. First, a student performs experiment five, crossing A and B flower types. This first cross yields flowers that are either heterogeneous Xx (type Ab) or homogeneous recessive xx (type B). The student then crosses children flowers with the same phenotype hoping to witness a solution. If the student crosses two type B children, he will perform experiment seven and need to try the other cross. If the student crosses two type Ab children and yields a garden with any type B flowers, he has performed experiment twelve or thirteen and found a solution. If the student is unlucky and performs experiment eleven and seven, he should start over. This example and others allow players to experiment freely, while the system compactly records their actions and can provide feedback for the players and educators.

V. FUTURE WORK

Experiment Centered Design provides a framework for assessing student learning and progress by tracking experiments completed by the students. Utilizing this design template allows other educational game developers to capture players experimenting within their game and map these actions to learning concepts. Future work with XCD has potential to utilize and build upon this framework.

Experiment Centered Design in Future Endeavors

Apart from *Radix*, a number of other educational games could make use of XCD. Future work could compare the pros and cons of utilizing ECD, Stealth Assessment, and XCD. Such a comparison could look at the ease of adoption, storage memory requirements, and versatility to rapid iteration.

Classification of Experiments in Experiment Centered Design

Experiment Centered Design extends ECD by giving specific form to the task, evidence, and student models. The ability to specify additional patterns in XCD may benefit developers by providing examples to guide their work. Categorizing experiments into particular families may help developers quickly fit an experiment into a particular XCD template.

This paper has begun to recognize differences in particular sets of experiments. One example is the application of XCD to closed- and open-response questions. Experiments that resemble open-response questions dynamically populate the database of tests performed by players. Experiments that resemble multiple-choice questions, however, can exhaust the list of tests performed by players and prepopulate the database.

The volume and surface area quest exposed another set of different experiments. The experiments developed by the game designer focused on the player submitting information for review. Players submitted a box with a particular length, width, and height to be judged. The experiments developed by the researcher focused on the player submitting information to create an artifact. The player input a length, width, and height to create a box. The flower breeding quest resembles the latter, where students input flowers to create a garden of more flowers.

With more examples, more patterns and classifications are likely to arise. By exploring these different classifications, the ability to apply XCD to new and different experiments will become easier and more efficient.

Social Experiments in Experiment Centered Design

One set of quests that has yet to be prototyped require multiple players interacting with each other as well as the environment. These "multiplayer" quests may require the players to perform series of experiments dependent on each other. Because the players share the set of experiments needed to solve the quest, XCD must analyze the union of the players' actions to interpret meaning. Extending XCD to handle this union may require additional object model sets, like "teams" that group multiple "users".

Presenting Feedback

Experiment Centered Design forms a model of a student based upon learning concepts. How this model is utilized and displayed depends on the audience. Students benefit from feedback that can immediately coach and correct their behavior. The feedback should highlight a student's successes while encouraging meditation on misconceptions. Most importantly, this feedback must balance immersing the student in the narrative of the game while urging the student to consider the real-world applications beyond.

Teachers benefit from the aggregation of student performances so they can teach to students' weaknesses. The feedback system must be a glanceable UI that brings students' misconceptions into focus. The game is a supplement to the teacher's curriculum, and treats quests with the same gravity as homework. For this reason, the teacher is more interested in the common struggles holding back the majority of his class instead of the individual actions of each player.

Researchers benefit from watching trends so they can identify patterns in students' actions. This feedback system may require filtering demographics or searching for specific patterns. Overall, the user interface must empower researchers to identify the strengths and weaknesses of the project through the performance of the students.

VI. CONCLUSION

The integration of learning assessment in educational open-world games is progressing to allow students a more immersive and supportive experience while they play. A variety of techniques, ranging from traditional to stealth assessment, are still being tested to balance guiding students and giving them freedom to explore. This paper described Experiment Centered Design (XCD), a framework for learning assessment that extends Evidence Centered Design and the vision of in-game, unobtrusive, assessment. The paper provided an object model that templates a game's design around experiments. *The Radix Endeavor* is a massively multiplayer online educational game that utilized XCD to capture students learning concepts through gameplay. Future work hopes to further refine the classification of experiments captured in XCD and display the data captured in meaningful interfaces for a variety of audiences.

WORKS CITED

Baker, R., & Clarke-Midura, J. (2013). Predicting Successful Inquiry Learning in a Virtual Performance Assessment for Science. *21st International Conference on User Modeling, Adaptaion, and Personlization.*

Behrens, J., Mislevy, R., Dicerbo, K., & Levy, R. (2012). Evidence Centered Design for Learning and Assessment in the Digital World. In M. Mayrath, J. Clarke-Midura, & D. Robinson, *Technology Based Assessment for 21st Century Skills: Theoretical and Practical Implications from Modern Research.* New York, NY: Springer-Verlag.

Blizzard Entertainment Inc. (2012, January). *What is World of Warcraft*. Retrieved January 2012, from World of Warcraft: http://us.battle.net/wow/en/game/guide/

Clarke-Midura, J. (2012, October 31). *Quest Template*. Retrieved from The Radix Endeavor Wiki: http://team.imaginationtoolbox.org/projects/stemmo/wiki/Quest_Template

Clarke-Midura, J. & Yudelson, M. (2013). Towards Identifying Students' Reasoning using Machine Learning. To appear in *Proceedings of the 16th International Conference on Artificial Intelligence and Education*.

Dieterle, E., & Clarke, J. (2006). Studying Situdated Learning in a Multi-User Virtual Environment. In *Encyclopedia of Multimedia Technology and Networking (2nd ed.).* Hershey, PA: Idea Group, Inc.

Gee, J. P. (2007). *What Video Games Have to Teach Us About Learning and Literacy.* New York, NY: Palgrave Macmillan.

Gillispie, L. (2012, October). *Welcome to the World of Warcraft in School Wiki*. Retrieved January 2012, from World of Warcraft in School: http://wowinschool.pbworks.com

Jackson, D. (Director). (2012, February 8). *Object Model Notation* [Motion Picture]. Retrieved from https://stellar.mit.edu/S/course/6/sp12/6.170/courseMaterial/topics/topic5/video/notation/notation.mp4

Klopfer, E. (2011, 8 11). Cosmos: Learning about Science, Technology, Engineering, and Mathematics (STEM) practices through a Massively Multiplayer Online Game (STEM.M.O.). Cambridge, Massachusetts, United States of America.

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003, May 12). *A Brief Introduction to Evidence-Centered Design.*

Quellmalz, E. S., Timms, M. J., Buckley, B. C., Davenport, J., Loveland, M., & Silberglitt, M. D. (2011). 21st Century Dynamic Assessment. In *Technology-Based Assessments for 21st Century Skills* (pp. 55-89). Information Age Publishing.

Sao Pedro, M., Baker, R., Gobert, J., Montalvo, O., & Nakama, A. (2013). Leveraging Machine-Learned Detectors of Systematic Inquiry Behavior to Estimate and Predict Transfer of Inquiry Skill. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 1-39.

Shute, V. J., Masduki, I., & Donmez, O. (2010). Conceptual Framework for Modeling, Assessing, and Supporting Competencies within Game Environments. Tallahassee, Florida, United States of America: Florida State University.

Shute, V., & Ventura, M. (2013). *Measuring and Supporting Learning in Games: Stealth Assessment.* Cambridge, MA: The MIT Press.

Steinkuehler, C.A. & Duncan, S.D (2008). Scientific habits of mind in virtual worlds. *Journal of Science Education and Technology*, 17(6), 530-543.

Williamson Shaffer, D., & Gee, J. (2012). The Right Kind of GATE: Computer Games and the Future of Assessment. In M. Mayrath, J. Clarke-Midura, & D. Robinson, *Technology Based Assessment for 21st Century Skills: Theoretical and Practical Implications from Modern Research.* New York, NY: Springer-Verlag.