Behavior Research Methods
2007, 39 (3), 635-650

Ensemble: A Web-based system for psychology survey and experiment management

Stefan T. Tomic and Petr Janata
University of California, Davis, California

We provide a description of *Ensemble*, a suite of Web-integrated modules for managing and analyzing data associated with psychology experiments in a small research lab. The system delivers interfaces via a Web browser for creating and presenting simple surveys without the need to author Web pages and with little or no programming effort. The surveys may be extended by selecting and presenting auditory and/or visual stimuli with MATLAB and Flash to enable a wide range of psychophysical and cognitive experiments which do not require the recording of precise reaction times. Additionally, one is provided with the ability to administer and present experiments remotely. The software technologies employed by the various modules of Ensemble are MySQL, PHP, MATLAB, and Flash. The code for Ensemble is open source and available to the public, so that its functions can be readily extended by users. We describe the architecture of the system, the functionality of each module, and provide basic examples of the interfaces.

Psychologists are often faced with numerous organizational issues when managing surveys and experiments. In social psychology research, many scales and questionnaires are employed as measurement instruments, while in cognitive psychology questionnaires are often used to collect metadata about participants in conjunction with discrimination judgments and reaction times. Such ancillary surveys are commonly administered on paper forms, thereby demanding subsequent data entry prior to data analysis. Sometimes, a few online forms are created using Web page authoring tools, but these do not provide for dynamically reassigning questions and sequences of forms or for a standard means of storing responses. Further, it is very difficult to dynamically associate stimuli with forms authored in this manner. Thus, administration of an otherwise efficient data collection mechanism becomes cumbersome and error prone.

In this paper, we describe *Ensemble*, a suite of interacting modules that provide tools to streamline and standardize the process of managing stimulus and survey presentation and analysis in the standard psychology laboratory. We should note that the intention of creating Ensemble was not to replace other experiment presentation and management software such as E-Prime (Psychology Software Tools, Pittsburgh, PA), Presentation (Neurobehavioral Systems, Albany, CA), or Experimetrix (Sona Systems, Tallinn, Estonia). For example, Ensemble does not provide resources for scheduling participants as does Experimetrix, nor for recording response times and providing immediate session statistics as does Presentation. Since we routinely use some of these other tools, we had no desire to reinvent the wheel. Rather, we designed Ensemble to fill a middle-ground that meets several research needs.

This flexible and extensive system was developed for use by a small research lab. Currently, a group of approximately six researchers is actively using an installation of Ensemble to create and administer experiments. MATLAB (MathWorks, Natick, MA) is tightly integrated into the framework, although labs that do not use MATLAB can also benefit from its use. Web-browser utilities are provided for creating questionnaires, organizing stimuli, and presenting the questionnaires and stimuli. Sequences of questions and forms are dynamically linked, providing the ability to quickly reorder or reassign them. Web forms may be authored without the need to learn HTML (the markup language used to author Web pages) or PHP (the scripting language employed in Ensemble for dynamic Web page processing). Interfaces are provided for analyzing data and dynamically selecting stimuli using MATLAB. A Flash player is integrated for presenting auditory or visual stimuli. Organization of questions, stimuli, and responses is formalized, reducing the time involved in initiating new experiments and facilitating collaboration between researchers. Questions, forms, and stimuli are used simultaneously by multiple experiments, reducing data redundancy. The data acquisition and analysis routines are integrated, easing the transition from one phase to the next in survey and experiment management. Finally, by incorporating a Web-based interface, Ensemble provides the capability to present surveys and experiments remotely, profoundly expanding the possible pool of participants.

P. Janata, pjanata@ucdavis.edu

Copyright 2007 Psychonomic Society, Inc.

ENSEMBLE OVERVIEW

The seven modules of Ensemble, indicated by the rectangles in Figure 1, interact with one another, the researcher, and the participant as outlined by the arrows in the figure. PhpMyAdmin (The phpMyAdmin Project, 2005), the Questionnaire Editor Interface (QEI), and the Questionnaire Presentation Interface (QPI) are Web interfaces that were written in PHP, a scripting language processed on the Web server for dynamic generation of Web pages. A MySQL database provides the central storage for Ensemble. PHP is capable of submitting and retrieving data from MySQL; this communication is denoted in the figure by the "PHP" label between the Web modules and MySQL. MATLAB communicates with the database through a MATLAB MySQL client. Below we provide an example of how the modules are typically used.

Question and Form Creation

The researcher opens the Questionnaire Editor Interface (QEI) with a Web browser. Questions authored with this interface are associated with forms, which are then assigned a logical order for presenting the survey. Instructions, question text, forms, and form sequences are submitted to the MySQL database through the QEI. Instructions assigned to the top of each form direct the participant regarding the set of questions on the form. For example, the sentence, "please give us some information on your musical background," instructs the participant that the following questions refer to his or her musical background. Questions are text excerpts that ask for a specific response, such as, "please list the musical instruments that you can play proficiently," or "what genres of music were you exposed to as a child?" Response types and constraints are also assigned to each question. For example, the response to the question "how many years of musical training have you had?" could be assigned an integer response. The researcher may then present the survey, or if desired, organize a set of stimuli to extend the survey to implement a psychophysics or cognitive experiment.

Stimulus Organization and Script Creation

Ensemble enhances basic presentation of surveys with the option to present stimuli via a Flash player. Stimulus files, e.g., mp3 files, are copied to the server's file system, and metadata associated with the stimuli, such as stimulus name, play duration, and file location, are submitted to the MySQL database with a command-line PHP script.

The researcher authors a MATLAB routine for dynamically selecting stimuli during survey presentation. A MATLAB routine, for example, might query the MySQL database for excerpts of music that were popular when the participant was between eleven and seventeen years of age, then randomly return a stimulus ID from the list each time the routine is called. Another, more simple MATLAB routine, might randomize a predefined set of stimuli, then return a stimulus ID from the set each time the routine is called until the list is exhausted. We should note that the stimulus selection logic for which we use MATLAB can be programmed in many other languages, including PHP. Thus, some Ensemble users may wish to modify the PHP scripts that handle form presentation, and optionally communication with MATLAB, with calls to functions implemented in a language of their choosing, e.g., Java. After the survey is organized, the researcher provides a Web link to the participant for launching the Questionnaire Presentation Interface (QPI).

Form Presentation

Once the QPI is launched, the participant interacts with a series of Web pages, called *QPI forms*. A series of initial forms usually only contain instructions and questions (e.g., to obtain background and demographic information). The participant enters his or her responses, which are submitted and stored in the database.

Stimulus Selection

Forms may also present stimuli prior to displaying questions. Forms call a MATLAB routine, such as the function described above that selects a piece of music based on the participant's age, for stimulus selection. The QPI communicates with MatRPC in order to run the MATLAB function and obtain a stimulus ID. MatRPC is a simple message passer to MATLAB engines, which are essentially MATLAB workspaces that are initialized, processed, and terminated in the background. MatRPC relays the result from the MATLAB function that was prepared by the researcher, in this case a stimulus ID, to the QPI.

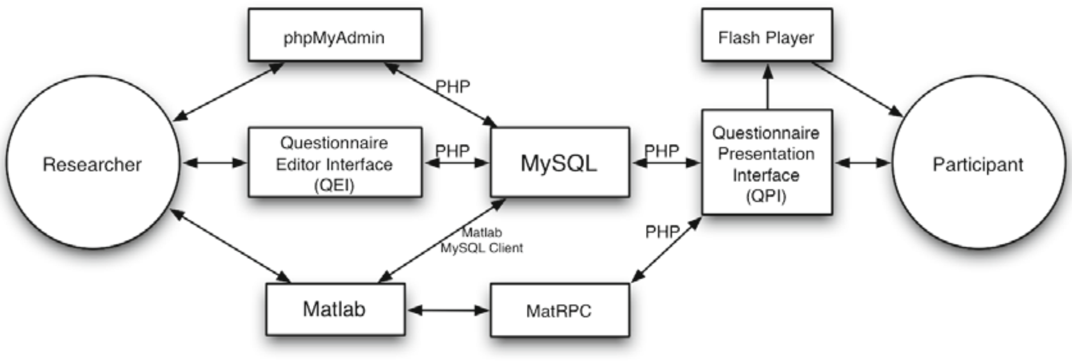


Figure 1. Diagram of Ensemble components. See text for more details regarding each component.

Stimulus Presentation

The QPI uses the stimulus ID to query the file location of the stimulus and passes this information to a Flash movie embedded in the browser. The Flash movie then opens the stimulus and presents it to the participant. Note that in the case of purely auditory stimuli, the Flash "movie" does not contain any video. After the Flash movie finishes presenting the stimulus, the QPI form presents its associated questions. For example, the QPI may present the question "How pleasing did you find the musical excerpt that just played?" after a musical stimulus. Although a timestamp is recorded when the form is submitted, this mechanism cannot record a precise reaction time of the response.

Data Analysis

After presenting QPI sessions, the researcher uses phpMyAdmin, a MySQL database administration utility, to query the data sets in order to make specialized selections of the responses or participant information. The researcher may also use MATLAB to retrieve responses from the database and analyze the data. If a different analysis application is desired, response and participant data may alternately be exported to a comma delimited text file through a provided Web utility. Once participant responses and demographic information are in the MATLAB workspace or in the exported text file, any number of analyses can be performed.

MODULE DESCRIPTIONS

Questionnaire Editor Interface (QEI)

The QEI consists of three Web-based utilities for creating questionnaires: Question Editor, Form Editor, and Experiment Editor. The utilities provide graphical interfaces for creating and managing questions, forms and experiments (collections of forms). The associated data are organized and submitted to the MySQL database by the utilities. Each utility consists of two frames (Figures 2, 3, and 4). A frame on the right is used for creating new questions, forms, or experiments; a second frame on the left is used to list and edit them.

The utilities require a username and password before being accessed. Ensemble is currently only configured for one researcher username and password. Since control over individual users is not provided, it is recommended that only a small number of researchers be given access to the administrative utilities. The researcher usually begins with the Question Editor to create and assign question text, data types for answers, allowed answer ranges, and answer format (e.g., pull down menu, blank text box, or radio buttons). Two types of multiple part questions are also supported. A *multiquestion* in Ensemble is used to implement series of questions that depend on one another and always appear together. *Multititle* questions display multiple response fields across columns in a tabular format.

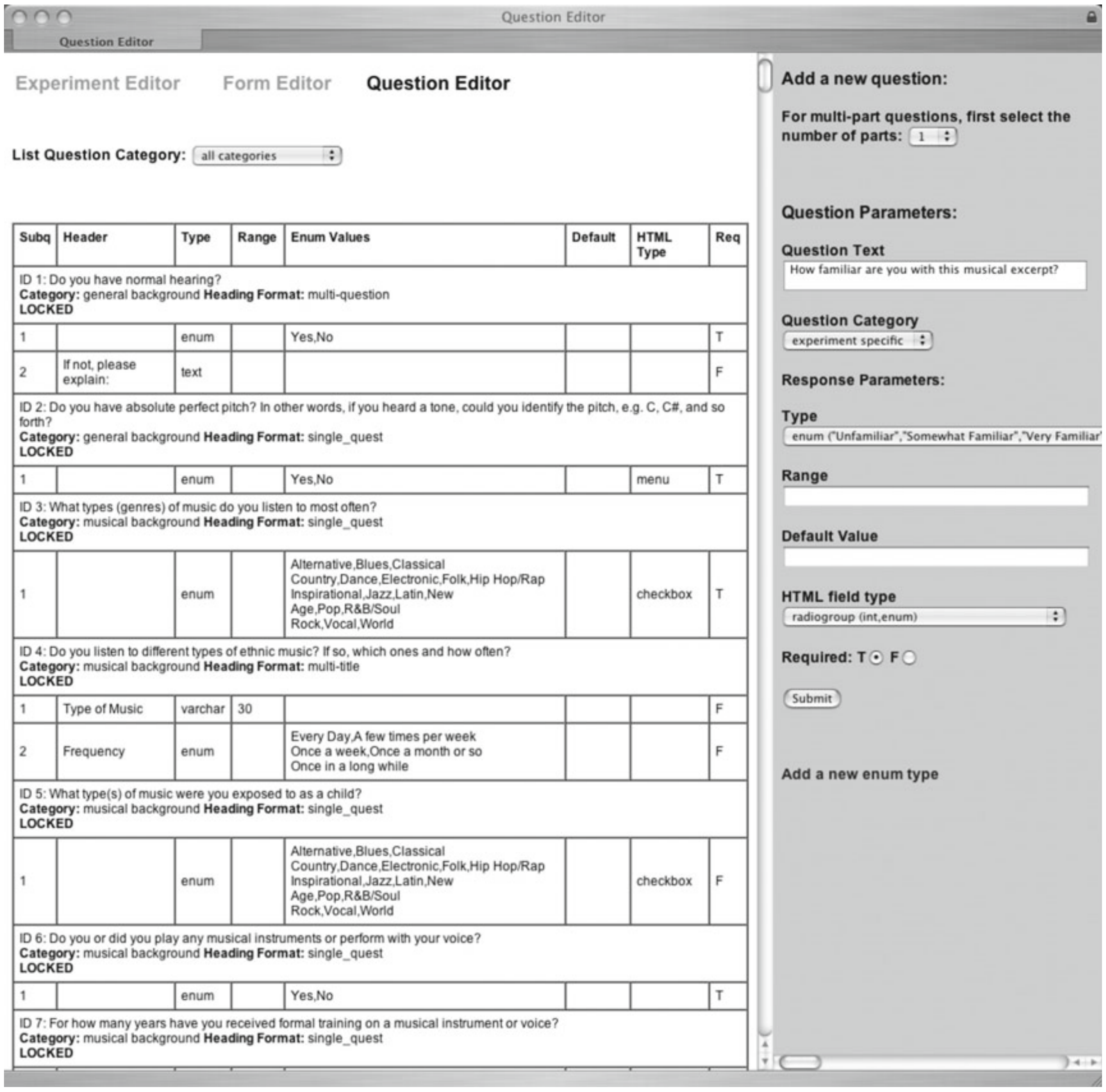
After the questions are authored, they are organized on forms using the Form Editor (Figure 3). The Form Editor provides interfaces for creating new forms, linking and removing questions from forms, and previewing forms. Multititle questions may be assigned multiple iterations in this editor for displaying multiple rows of response fields during experiment presentation. A "visit once" checkbox in the Form Editor provides the means to set a flag that ensures that the participant does not revisit the form if he or she has seen it during a previous experiment or session. A checkbox is also provided to lock the form so that it isn't inadvertently edited by another researcher who has access to the editors. One would need to intentionally toggle the flag in order to make changes to the form. Once the researcher has finished creating the forms, he or she arranges the logical order of the forms in the Experiment Editor.

The Experiment Editor (Figure 4) provides an interface for adding, removing, and prescribing the order of forms in an experiment. A utility for copying existing experiments is also provided, in the event that a researcher wishes to make modifications to an existing experiment and keep the old version intact. A description field for the experiment may be used to record the differences in experiment versions. A form handler, which provides the display and control logic during experiment presentation, is also assigned to each form using this interface. When an experiment is created, two forms are automatically added: one form starts the session and another ends the session. The form handlers linked to these forms cannot be changed, since they start the experiment by initializing, and end the session by deleting, session variables. However, any form created in the Form Editor can be associated with these form handlers in order to vary the text that greets the participant and concludes the session.

One may place sequences of forms in loops using the Experiment Editor. This is useful for the association of a fixed set of forms with a series of stimuli. The number of iterations and the location of the beginning and end of the loop are set in the Experiment Editor. Although the loop is easily established in the Experiment Editor, the logic for varying the stimuli between loop iterations or for breaking out of the loop in the case that the number of stimuli is variable, must be supplied by specialized code for the experiment. Ensemble was designed so that a researcher could write a function in MATLAB for this purpose, which would be called through MatRPC. Alternately, PHP code could be placed in a form handler to perform this logic (see discussion on form handlers in the QPI section).

While no graphical tools have yet been incorporated into the QEI for submitting stimulus information to the database, a PHP command-line script is available for submitting this information. Typically, one places all of the stimuli, which may be audio or video files, in a directory, opens the script and changes a few variables corresponding to directory paths and description fields of the stimuli, then runs the script for submission.

Finally, the Experiment Editor provides an interface for creating Ensemble tickets. Tickets are Web links which authorize a user to initiate a session for a particular experiment. The Web links are in the form of standard Web addresses with unique codes appended to them. Two types of tickets are available: master and user tickets. Master tickets can be used more than once whereas user tickets can only be used for a single session. This provides a mechanism for researchers to reuse the same ticket when



Question Editor

Experiment Editor Form Editor **Question Editor**

List Question Category: all categories

Subq	Header	Type	Range	Enum Values	Default	HTML Type	Req
ID 1: Do you have normal hearing? **Category:** general background **Heading Format:** multi-question **LOCKED**							
1		enum		Yes,No			T
2	If not, please explain:	text					F
ID 2: Do you have absolute perfect pitch? In other words, if you heard a tone, could you identify the pitch, e.g. C, C#, and so forth? **Category:** general background **Heading Format:** single_quest **LOCKED**							
1		enum		Yes,No		menu	T
ID 3: What types (genres) of music do you listen to most often? **Category:** musical background **Heading Format:** single_quest **LOCKED**							
1		enum		Alternative,Blues,Classical Country,Dance,Electronic,Folk,Hip Hop/Rap Inspirational,Jazz,Latin,New Age,Pop,R&B/Soul Rock,Vocal,World		checkbox	T
ID 4: Do you listen to different types of ethnic music? If so, which ones and how often? **Category:** musical background **Heading Format:** multi-title **LOCKED**							
1	Type of Music	varchar	30				F
2	Frequency	enum		Every Day,A few times per week Once a week,Once a month or so Once in a long while			F
ID 5: What type(s) of music were you exposed to as a child? **Category:** musical background **Heading Format:** single_quest **LOCKED**							
1		enum		Alternative,Blues,Classical Country,Dance,Electronic,Folk,Hip Hop/Rap Inspirational,Jazz,Latin,New Age,Pop,R&B/Soul Rock,Vocal,World		checkbox	F
ID 6: Do you or did you play any musical instruments or perform with your voice? **Category:** musical background **Heading Format:** single_quest **LOCKED**							
1		enum		Yes,No			T
ID 7: For how many years have you received formal training on a musical instrument or voice? **Category:** musical background **Heading Format:** single_quest **LOCKED**							

Add a new question:

For multi-part questions, first select the number of parts: 1

Question Parameters:

Question Text
How familiar are you with this musical excerpt?

Question Category
experiment specific

Response Parameters:

Type
enum ("Unfamiliar","Somewhat Familiar","Very Familiar"

Range

Default Value

HTML field type
radiogroup (int,enum)

Required: [✓ T] F

Submit

Add a new enum type

Figure 2. QEI: Question Editor.

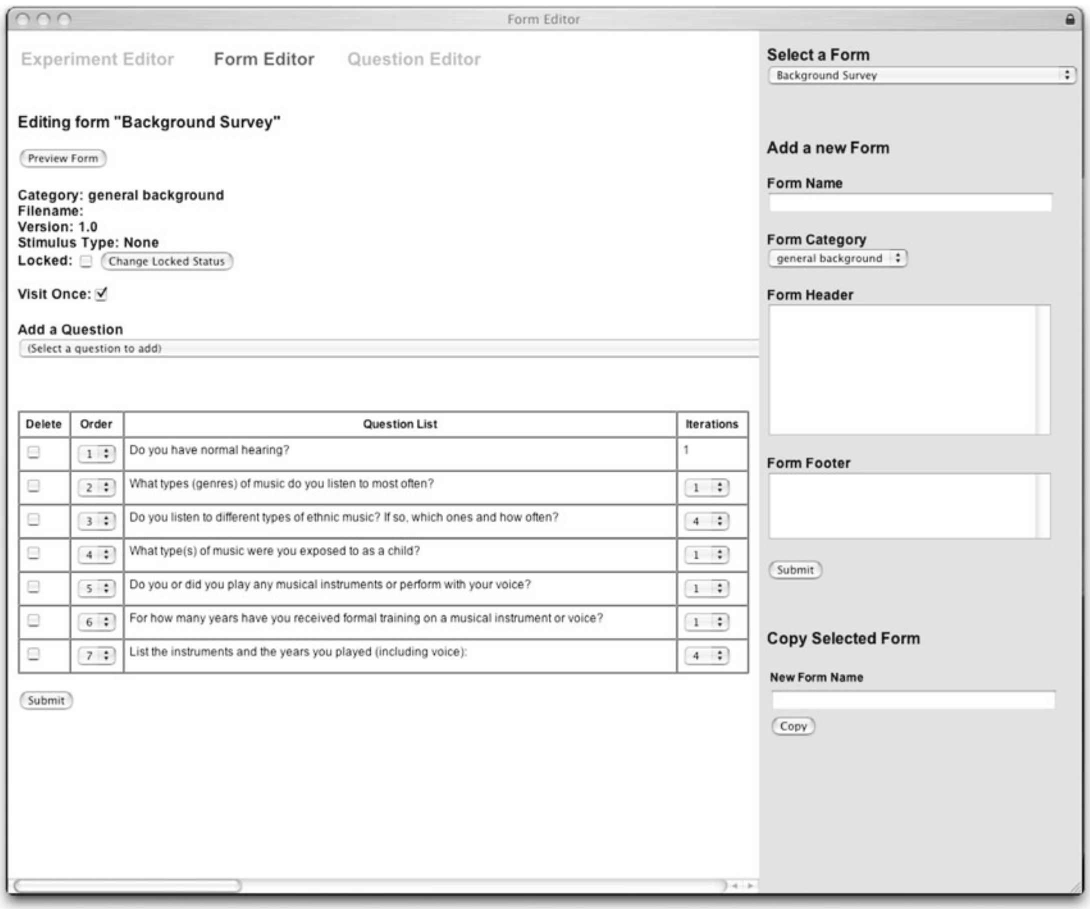
administering an experiment in a controlled environment. User tickets are generally used when providing links to remote participants. The single use restriction for user tickets prevents participants from running the experiment multiple times or advertising the link to others. If desired, expiration dates can be set for master and user tickets after which they can no longer be used. Note that master tickets may still be given to remote participants. There would simply be no limit to the number of sessions that could be launched from this ticket. In our lab, we generate master tickets without expiration dates and user tickets with expiration dates for remote participants in order to ensure that the experiment is not executed after the official period of data collection.

The ticket ID, and therefore also the ticket type, used with each session is recorded. This provides a mechanism for the researcher to mask out specific sets of sessions in the analysis phase. For example, one can run separate analyses on remote participants and those that participated in a controlled environment by selecting for a ticket type, or for a particular master ticket.

Questionnaire Presentation Interface (QPI)

The Questionnaire Presentation Interface (QPI) presents the questions, forms and stimuli created and organized with the QEI. It is launched with a master or user ticket. The participant is presented with a page that starts the experiment. The form sequence specified in the QEI Experiment Editor and stored in the MySQL database follows.

Each Web page presented by the QPI is called a form in Ensemble. Forms can be either of a *trial* or *nontrial* type. While nontrial forms are capable of presenting stimuli, their primary function is presenting questions and their association to stimuli is entirely established through form handlers (see below). Even though nontrial forms do not link directly to stimuli in the database, those that present stimuli do submit the stimulus ID with the responses. Trial



Editing form "Background Survey"

Preview Form

Category: general background
Filename:
Version: 1.0
Stimulus Type: None
Locked: ☐ Change Locked Status

Visit Once: ✓

Add a Question
(Select a question to add)

Delete	Order	Question List	Iterations
☐	1	Do you have normal hearing?	1
☐	2	What types (genres) of music do you listen to most often?	1
☐	3	Do you listen to different types of ethnic music? If so, which ones and how often?	4
☐	4	What type(s) of music were you exposed to as a child?	1
☐	5	Do you or did you play any musical instruments or perform with your voice?	1
☐	6	For how many years have you received formal training on a musical instrument or voice?	1
☐	7	List the instruments and the years you played (including voice):	4

Submit

Select a Form
Background Survey

Add a new Form
Form Name
Form Category
general background
Form Header
Form Footer
Submit

Copy Selected Form
New Form Name
Copy

Figure 3. QEI: Form Editor.

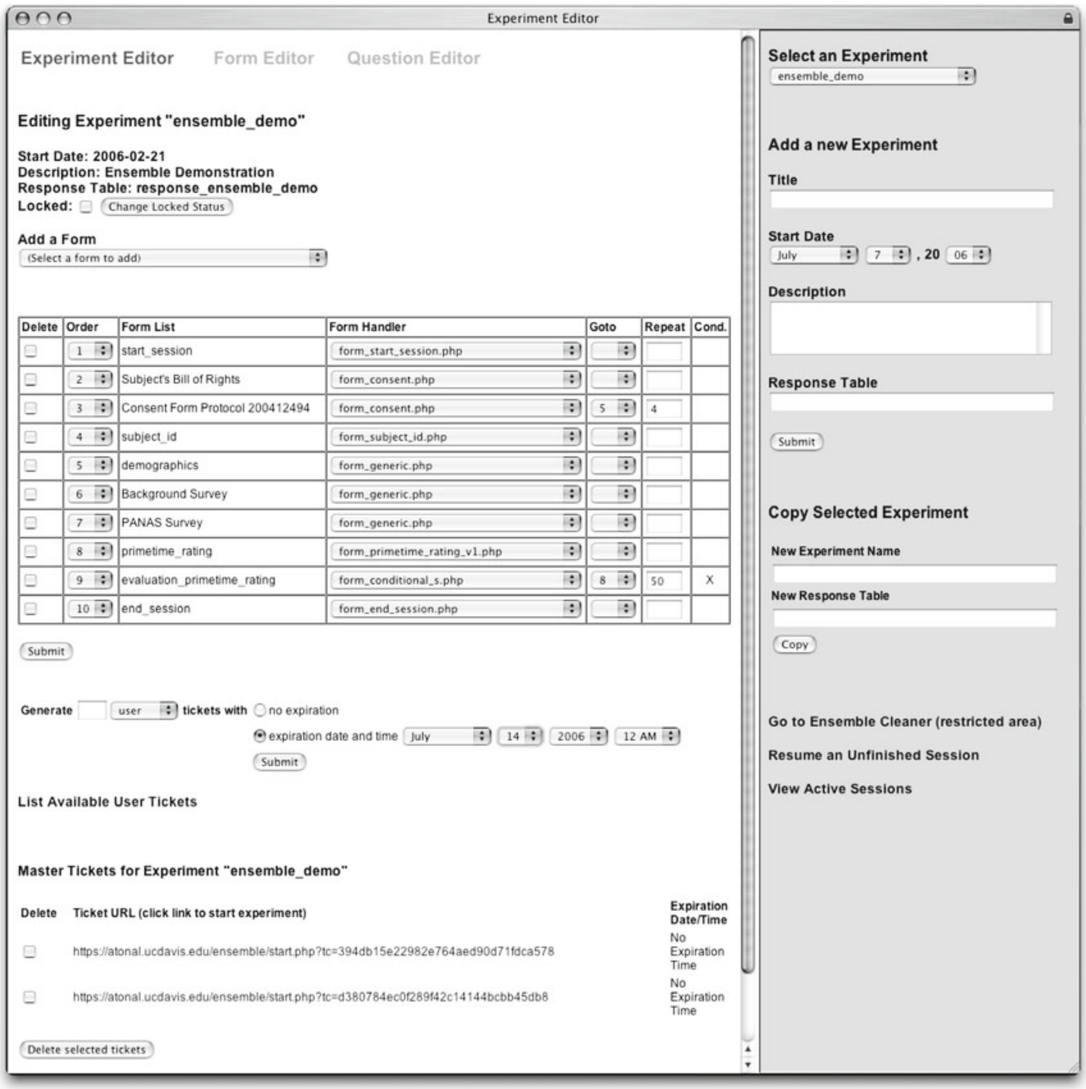
forms, on the other hand, link to trials in the MySQL database. Each trial is associated with a specific question, a correct answer, and a single stimulus or a pair of stimuli. Trials provide a mechanism to organize possible combinations of questions and stimuli, which may be randomized or presented in order depending on the form handler used or the MATLAB function called through MatRPC. Questions, in addition to stimuli, vary on a trial form whereas a nontrial form always presents a fixed set of questions.

The arrangement of questions on a form, presentation of associated stimuli, and writing of responses to the database are accomplished with form handlers. A form handler is assigned to each form, prior to running any QPI sessions, in the Experiment Editor. Most forms use a generic form handler that displays questions and response entry fields without playing stimuli or performing specialized logic. If a form's occurrence is conditional on previous responses, the form is trial-based, stimuli need to be presented, or a MATLAB function must be called, specialized form handlers are assigned. For example, a specialized form handler might call a MATLAB function through MatRPC to obtain a stimulus ID, then play the stimulus before presenting questions associated with the form. Typically, specialized form handlers differ from the generic form handler by only a few lines of code, and once created, specialized form handlers can be used in many different experiments with minimal modification. Thus, a relatively small number of form handlers can be used to manage a very broad range of surveys and experiments with no, or only minimal, programming effort by the end-user.

A basic flowchart for a single QPI form handler is illustrated in Figure 5. Each of the boxes correspond to a PHP script called by the form handler. In order to change the logic of the form handler, one only needs to add or remove the line of code that calls the PHP script.

Forms are designated as trial or nontrial forms by the form handlers chosen. The two types of forms are handled very similarly and differ only in the way that information about them is queried from the database. The former are linked directly to questions while the latter are linked to a series of trials, each of which is linked to a question and a stimulus or pair of stimuli. A trial form is usually placed in a loop in the experiment so that the form can progress through the trials by querying each successive trial through its form handler.

Conditional forms, which are forms that are only presented if specific responses were made for previous questions, use a PHP script called by the form handler to check for those responses. A condition code, which describes the



Editing Experiment "ensemble_demo"

Start Date: 2006-02-21
Description: Ensemble Demonstration
Response Table: response_ensemble_demo
Locked:

Add a Form
(Select a form to add)

Delete	Order	Form List	Form Handler	Goto	Repeat	Cond.
	1	start_session	form_start_session.php			
	2	Subject's Bill of Rights	form_consent.php			
	3	Consent Form Protocol 200412494	form_consent.php	5	4	
	4	subject_id	form_subject_id.php			
	5	demographics	form_generic.php			
	6	Background Survey	form_generic.php			
	7	PANAS Survey	form_generic.php			
	8	primetime_rating	form_primetime_rating_v1.php			
	9	evaluation_primetime_rating	form_conditional_s.php	8	50	X
	10	end_session	form_end_session.php			

Generate user tickets with no expiration [✓ expiration date and time] July 14 2006 12 AM

List Available User Tickets

Master Tickets for Experiment "ensemble_demo"

Delete	Ticket URL (click link to start experiment)	Expiration Date/Time
	https://atonal.ucdavis.edu/ensemble/start.php?tc=394db15e22982e764aed90d71fdca578	No Expiration Time
	https://atonal.ucdavis.edu/ensemble/start.php?tc=d380784ec0f289f42c14144bcbb45db8	No Expiration Time

Select an Experiment
ensemble_demo

Add a new Experiment

Title

Start Date
July 7 , 20 06

Description

Response Table

Copy Selected Experiment

New Experiment Name

New Response Table

Go to Ensemble Cleaner (restricted area)

Resume an Unfinished Session

View Active Sessions

Figure 4. QEI: Experiment Editor.

responses on previous forms to be matched, is associated with the form in the MySQL database and is referenced in this process.

Form handlers provide functions to Ensemble that extend beyond the simple survey format. They were designed for easy customization and for providing communication with MatRPC. This allows a lab to take advantage of the capabilities of MATLAB to develop specialized logic for an experiment. If one wishes not to use MATLAB code, it is also possible to provide controls of stimuli, trials and loops exclusively within the PHP code of the form handler. This would require some familiarity with PHP, however.

Responses are checked for validity before submission to the MySQL database. If responses are of incorrect data types (e.g., alphabetical characters where an integer is required), incorrect ranges (e.g., a value of 30 where a number between 1 through 10 is required), or are blank for required questions, the participant is sent back to the form with an error message so that he or she can correct them.

Online consent forms, if used in lieu of paper consent forms in the experiment, use a specialized form handler which displays the consent form text and an option to agree or disagree. A statement on the form informs the participant that selecting "agree" constitutes his or her electronic signature. If the participant disagrees, the form handler terminates the session immediately with a message explaining that the experiment cannot continue without the participant's consent. Consent forms precede the subject_id form, as participants must provide consent before the researcher can collect information about them. The consent form responses are recorded in the response table for the experiment, and are linked to a temporary subject ID until a permanent ID is generated. The online consent forms used in our lab have been approved by the UC Davis Institutional Review Board (IRB) and replace paper forms we previously employed.

The participant is then directed to the subject_id form which asks for his or her name and birthday. This is dis-

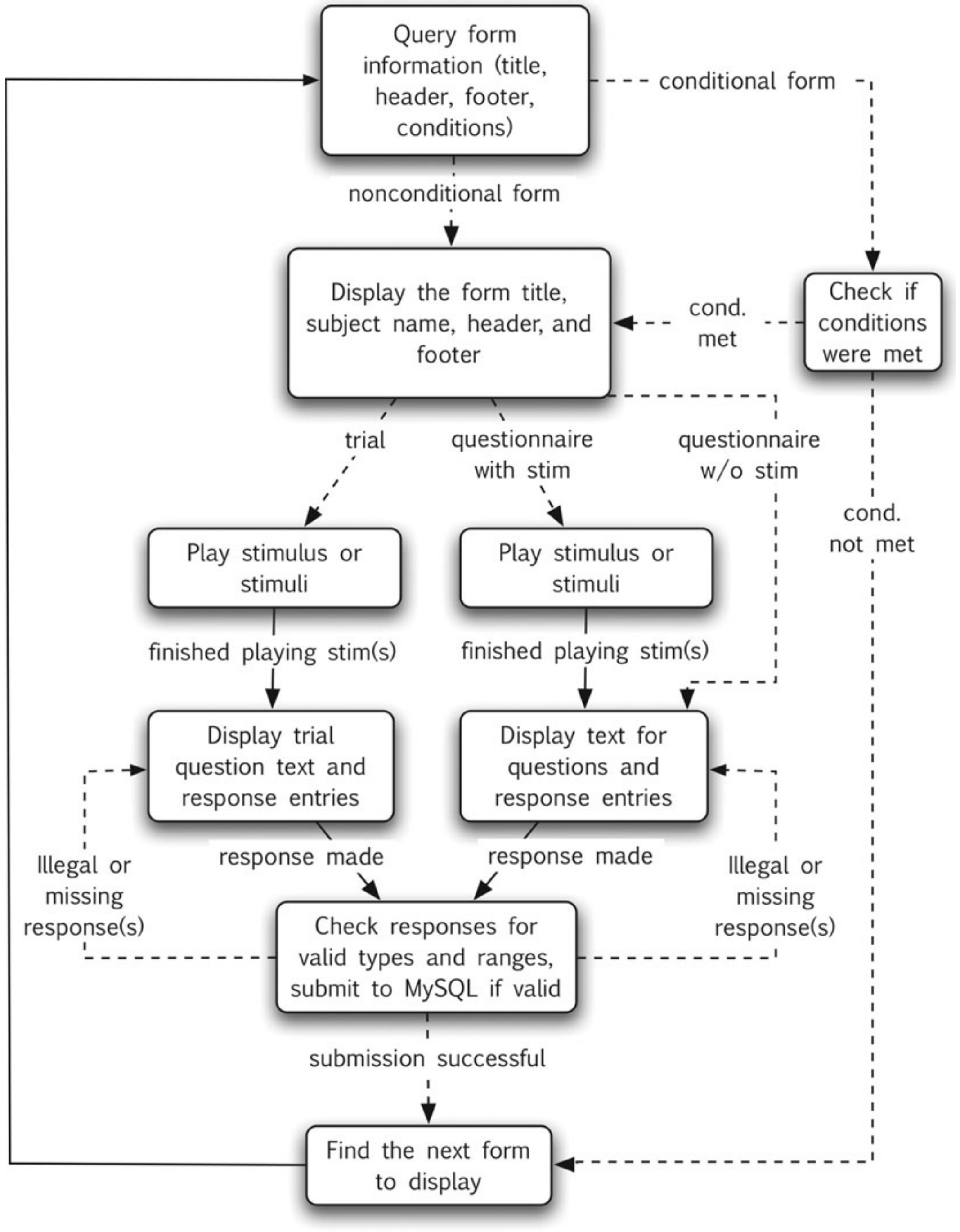

Query form information (title, header, footer, conditions)
conditional form
nonconditional form
Display the form title, subject name, header, and footer
cond. met
Check if conditions were met
trial
questionnaire with stim
questionnaire w/o stim
cond. not met
Play stimulus or stimuli
Play stimulus or stimuli
finished playing stim(s)
finished playing stim(s)
Display trial question text and response entries
Display text for questions and response entries
Illegal or missing response(s)
response made
response made
Illegal or missing response(s)
Check responses for valid types and ranges, submit to MySQL if valid
submission successful
Find the next form to display

Figure 5. QPI Form Handler flowchart.

played by a subject_id form handler, which generates a subject ID based on this information. The participant's name and birthday are stored in a "subject" table on the database and any other information provided by the participant is linked to the ID. All subsequent responses are linked to one's subject ID and consent responses are updated so that they link to the subject ID. Alternately, a form which generates an anonymous subject ID can be used instead of the subject_id form, if one wishes not to collect any identifying information about the participant. Access to the subject table can be restricted to a subset of authorized users, adding a further measure of privacy protection. We add a demographics form after the subject ID form. This form is required for reporting demographic information of our participant pool in experiments funded by the National Institutes of Health (NIH) in the United States (Form PHS 398, NIH). Responses to this form are optional for participants. Prior to using Ensemble, we collected this data on paper forms. Specific forms for the experiment follow.

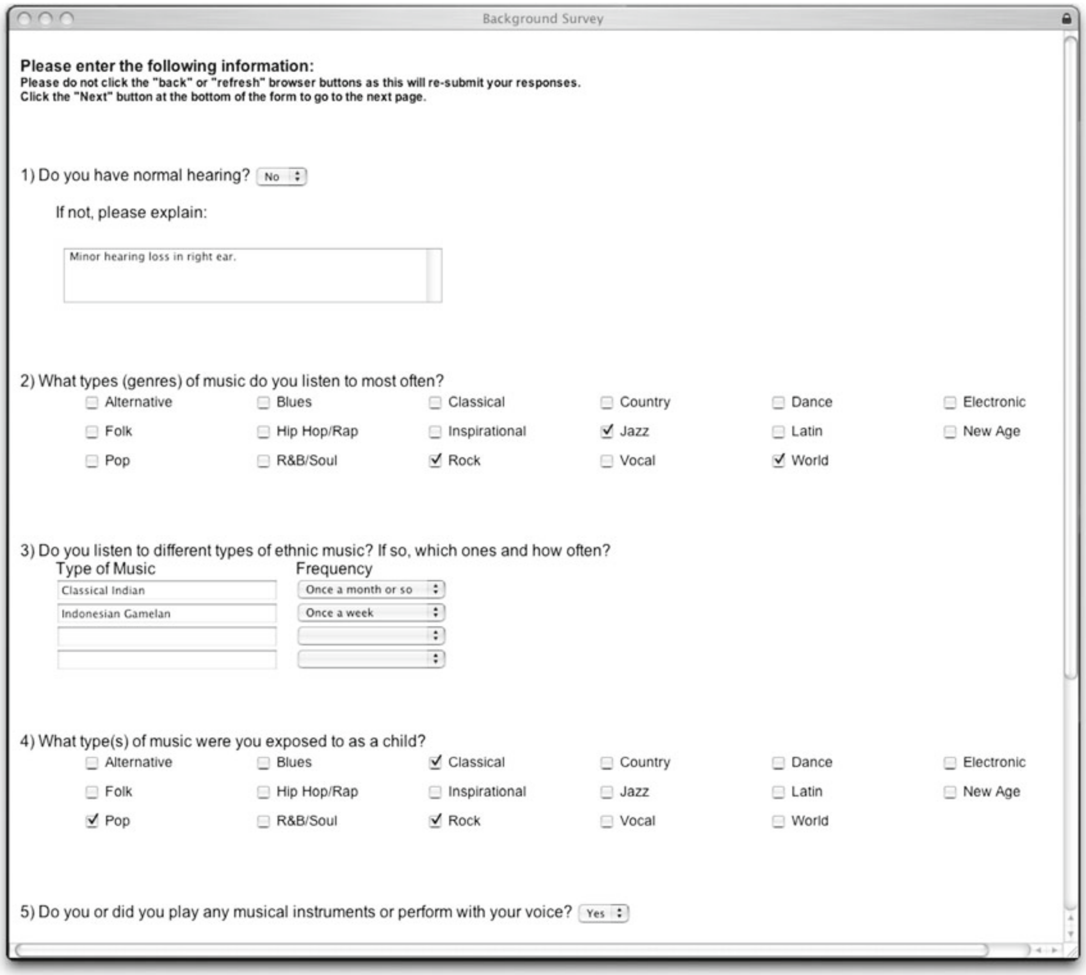
A form rendered by the QPI is shown in Figure 6. A few question types are illustrated. Question 1 is implemented as a multiquestion so that the two questions always appear together. Question 3 is a multititle question with four iterations. Questions 2, 4, and 5 are questions linked to enumerated types defined by the researcher. Question 5 presents a pulldown menu instead of checkboxes, a formatting choice that was specified in the QEI. In addition to providing different appearances, answer display formats also provide different capabilities for multiple choice answers. Pulldown menus and radiobuttons only allow a single answer at a time while checkboxes (Questions 2 and 4) provide for multiple simultaneous answers.

If the Web browser crashes or is accidentally closed during a QPI session, a utility is available in the QEI for resuming the session. This is also useful for presenting two sets of forms in the same experimental session but from different locations. The browser may be closed after the initial set of forms. The second set of forms, which are associated with the same experiment, can be accessed on an altogether different computer by resuming the session through the QEI. After all of the QPI forms have been presented to the participant, the end_session form clears the session variables and concludes the session.

MySQL

The MySQL database is central to the framework of Ensemble as it provides a mechanism for storing, submitting, and retrieving data. MySQL databases are organized into a series of tables. A custom database schema was developed for organizing the tables used in Ensemble. The structures of the experiment, form, trial, question, and data_format tables, which constitute a partial schema of the MySQL database, are illustrated in Figure 7. The illustration was produced by importing the database schema into a MySQL compatible tool called DBDesigner (Zinner, 2003). The ability to import the schema and produce the image with relative ease using the open source third-party application highlights one of the benefits of using ubiquitous open-source components.

Basic tables. Each table in the schema contains one or more fields. For example, the experiment table con-



Please enter the following information:
Please do not click the "back" or "refresh" browser buttons as this will re-submit your responses.
Click the "Next" button at the bottom of the form to go to the next page.

1) Do you have normal hearing? No

If not, please explain:

Minor hearing loss in right ear.

2) What types (genres) of music do you listen to most often?

☐ Alternative	☐ Blues	☐ Classical	☐ Country	☐ Dance	☐ Electronic
☐ Folk	☐ Hip Hop/Rap	☐ Inspirational	✓ Jazz	☐ Latin	☐ New Age
☐ Pop	☐ R&B/Soul	✓ Rock	☐ Vocal	✓ World	

3) Do you listen to different types of ethnic music? If so, which ones and how often?

Type of Music	Frequency
Classical Indian	Once a month or so
Indonesian Gamelan	Once a week

4) What type(s) of music were you exposed to as a child?

☐ Alternative	☐ Blues	✓ Classical	☐ Country	☐ Dance	☐ Electronic
☐ Folk	☐ Hip Hop/Rap	☐ Inspirational	☐ Jazz	☐ Latin	☐ New Age
✓ Pop	☐ R&B/Soul	✓ Rock	☐ Vocal	☐ World	

5) Do you or did you play any musical instruments or perform with your voice? Yes

Figure 6. Form rendered by the QPI.

tains the experiment_id, start_date, experiment_title, experiment_description, and response_table fields. Records contain the associated field values of data instances in the table. Fields can be conceptualized as table columns, while records can be thought of as rows (Table 1) akin to how one often organizes a table in a spreadsheet program such as Excel.

Table 1
Experiment Table Records

experiment_id	start_date	experiment_title	response_table
1	1/1/05	Groove	response_groove
2	8/10/05	Autobio fmri	response_autobio_fmri
3	1/5/06	Pop-out	response_popout

Each record in Table 1 contains the associated field values for a single experiment. A record is identified by its primary key, which is a field or series of fields that must contain a unique value or combination of values. Primary keys are commonly single integer fields that increment with each successive record, e.g., the experiment_id field in the experiment table. This type of primary key, often referred to as the ID field of a table, is a straightforward and robust method for ensuring uniqueness. A primary key composed of multiple fields is referred to as a composite key. In this situation, the group of fields used as a primary key constitute a unique combination of values. For example, the experiment_x_form table (bottom-left of Figure 7), which is a special table called a junction table (discussed in a following subsection), uses the experiment_id, form_id, and form_order fields as a composite key.

The organization of the QEI into experiment, form, and question editor interfaces mirrors the organization of tables in the database schema in the sense that experiments link to forms and forms link to questions. Each experiment is associated with a series of forms stored in the form table. The form table contains fields for the form ID, form name, form category, header text, footer text, version number, condition code, and a flag that designates the form as a "visit once" type. The condition code is used for conditional forms and specifies the set of responses to questions on previously presented forms which must have been made for a conditional form to be displayed during a QPI session. Each form is associated with a series of questions stored in the question table. Forms may alternatively link to a series of trials in the trial table instead of linking directly to questions.

The question table stores the question text and attributes, such as the heading format and the question category. The heading format designates the question type as a multiquestion, multititle, or single question. A lock flag is set when the question has been used in an experiment. This prohibits further editing of the question in the QEI. The data_format table stores data type constraints for responses to the questions.

Table relationships. The associations between tables are established within the database schema through rela-

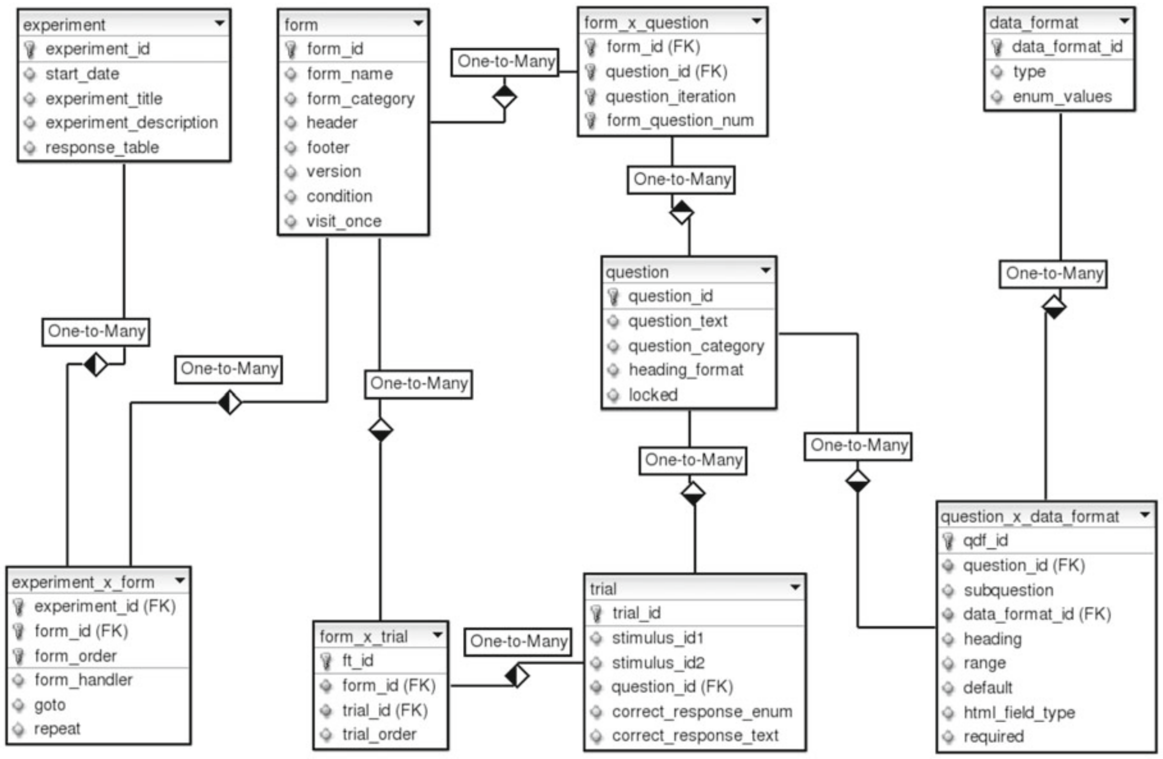


Figure 7. Experiment, form, trial, question, and data format table schema. Each of the boxes containing key and diamond symbols represents a table in the database. The name of the table is given at the top of the box, and the field names follow below. The key symbols indicate the field(s) that comprise the table's primary key which uniquely identifies a record in the table. Foreign keys (primary keys from other tables) are denoted by FK. This graphic was plotted with DBDesigner 4.

tionships. A relationship links records together. For example, multiple question records link to a single form record, and multiple form records link to a single experiment record in order to establish the sequences of a QPI session. Relationships are usually between records of two different tables, but may also be established between records within a single table. The following discussion includes only relationships between two tables since single table relationships are not used in Ensemble. Relationships are established through the use of foreign keys, which are fields in one table that reference primary key fields of another table by duplicating their values.

Relationship types between two tables are described as one-to-one, one-to-many, and many-to-many. A one-to-many relationship exists when one record from a table is linked to multiple records of a second table, but every record in the second table is associated with only one record from the first. An example of a one-to-many relationship is illustrated between the question table and trial table in Figure 7. The question_id field in the trial table is a foreign key that refers to the question_id primary key field in the question table. This provides the means to link one or more trials to a question. For example, the inquiry, "Do you find this excerpt pleasing, displeasing, or neutral/mixed?" may be associated with a number of trials, each of which links to a different stimulus.

Usually, one does not need to establish a one-to-one relationship (a reference between a single record from one table and a single record from another table) since all the data in the relationship can often be stored in a single table. Many-to-many relationships need special treatment and are established by the use of junction tables.

Junction tables. Junction tables link two tables that have a many-to-many relationship. They are identified in Ensemble by the name of the first table in the many-to-many relationship, followed by the string "_x_" and the name of the second table. A many-to-many relationship exists when a single record in each of two tables has the possibility of linking to multiple records in the other table, e.g., an experiment is associated with one or more forms, and each form is associated with one or more experiments. Each of the two tables in a many-to-many relationship is linked to a junction table with a one-to-many relationship. In addition to establishing many-to-many relationships, junction tables may also store information specific to the relationship.

Consider, for example, the many-to-many relationship between the experiment table and the form table. The experiment and form tables each link to the experiment_x_form junction table through a one-to-many relationship, i.e., each record in the experiment and form tables links to multiple records in the experiment_x_form table. The junction table provides the ability to link an experiment record to multiple form records in order to establish the form sequence of the experiment. Forms may be linked to multiple experiments so that they can be reused among experiments. In addition to establishing the many-to-many relationship, the experiment_x_form table prescribes the order of the forms of the experiment by the form_order field and stores the form handler and looping information for the form. By storing this information in the junction table, forms that are used in more than one experiment may occur in a different order, and be associated with different form handlers and looping patterns.

Forms are linked to a series of questions through the form_x_question junction table. The form_x_question table also stores fields for the number of question iterations and the order of occurrence of the question. By storing this information in the junction table, a question may occur in a different order and have a different number of iterations for each form which uses it.

Forms which use trials do not link directly to questions and instead link to a series of trials through the form_x_trial table. Each record in the trial table in turn links to one or two stimuli in the stimulus table, a question from the question table, and contains a field for the correct response (response_enum or response_text). This organization allows for 2-alternative forced-choice experiments to be realized in Ensemble. A QPI form handler handles the progression of trials for the form, which can be random, algorithmic, or sequential. Such configurations are suitable for psychophysics experiments, in which participants must make subtle auditory or visual discriminations, or in behavioral experiments such as categorization tasks or memory judgments.

Finally, questions are associated with data formats, which are used as constraints of the responses to the questions, through the question_x_data_format table. Linking of a single question to multiple data formats provides the implementation for multiple part questions by associating a data type with each answer. A *subquestion* in Ensemble refers to a part of a multiple part question, and a subquestion number refers to its order. The question text, default answer, and subquestion (number) fields are stored in the question_x_data_format junction table since they describe the relationship between a specific question and a data format. The link between a single data format record and multiple questions, on the other hand, provides the ability to reuse, and thereby standardize, data formats.

Data formats. There are 10 predefined data formats, or types, in Ensemble: varchar, text, int16, int32, int64, double, date, time, year, and null. The numeric and string types are compatible with both MySQL data types and MATLAB while the date and time formats follow MySQL conventions and can be handled in MATLAB as formatted strings. Varchar is a string type with a specified maximum string length limit while text handles long string lengths up to 65,536 characters. Int16, int32, and int64 are simply 16-, 32-, and 64-bit signed integers as the names suggest. Double is a double-precision (64-bit) floating-point type. Null is used for questions which do not provide for a response, and usually occur in subquestions. For example, a multiple part question which uses a null type for the first subquestion is shown in Figure 8. The first subquestion simply displays text for the general question while the following five subquestions are words or phrases which require specific responses. Thus, the phrase "writing" is the second subquestion.

The researcher may add a large number of enumerated types.[1] Enumerated types limit the possible answers to a fixed set of options. For example, the set of words "displeasing," "neutral," and "pleasing" may constitute an enumerated type. Responses to questions that use the enumerated type are selected from the set of words. Enumerated type definitions are stored as a set of strings in the enum_values field of the data_format table. If a numeric enumerated type is desired, one may define each element of the set to be a number stored as a string. The primary advantage of enumerated types is that they provide a mechanism for various "textual" and categorical responses while avoiding the problems of typographical errors, differences in case, and use of synonyms which are encountered when participants are free to type in a response.

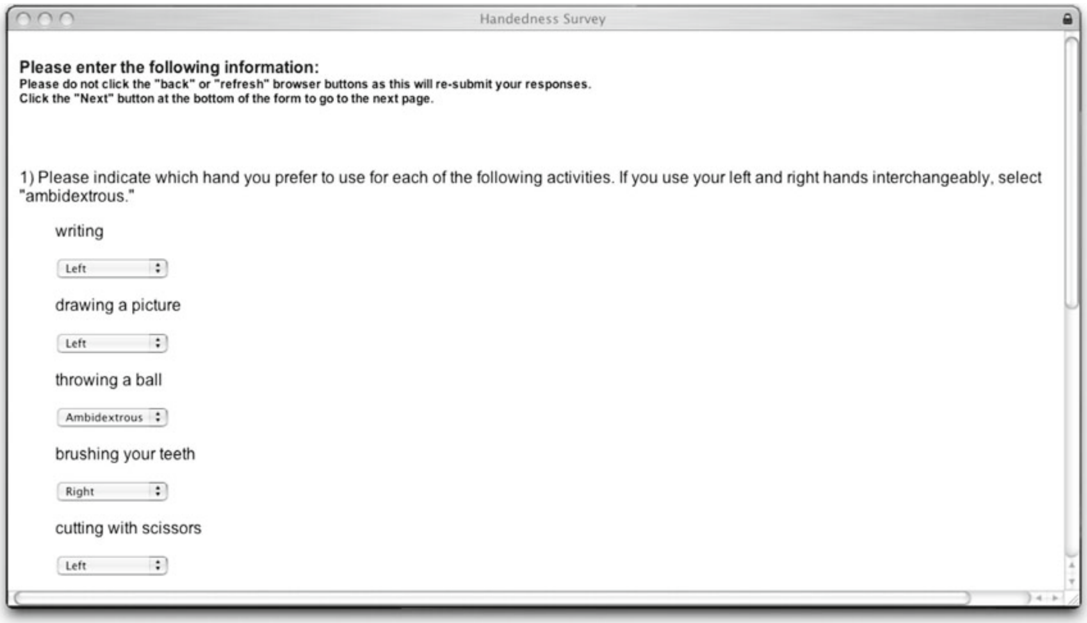
Stimuli. Stimuli are stored in the file system of the server. Information about the stimuli is stored in the stimulus table (Figure 9). Most of the stimulus fields correspond to tags associated with MP3 stimuli. Importing and display of audio stimuli is largely compatible with iTunes and playlists created in iTunes. The fields in the stimulus table are generally useful for musical audio files. With the exception of stimulus_id and location fields, fields can be added or removed from the stimulus table without affecting the integrity of the system. The location field specifies the directory path and file name of the stimulus.

Stimuli may be referenced in an experiment through their attributes. An attribute is a descriptor for categorizing a group of stimuli. Examples of attribute types commonly used in our lab are musical genres or sets of specific stimuli associated with particular experiments. Stimuli are linked to attributes through the stimulus_x_attribute junction table in order to link multiple attributes to a single stimulus and link multiple stimuli to a single attribute. The attribute_x_attribute table provides the ability to create a hierarchy of attributes through an attribute parent and child scheme by linking attributes to each other. For example, one hierarchy might specify the genre attribute "rock music" which has as a few of its children attributes, or subgenres, "alternative rock," "punk rock," "heavy metal," and "60's psychedelic rock." During experiment presentation, the QPI may then choose a set of stimuli based on an attribute or group of attributes.

Response tables. While the tables discussed so far handle data for multiple experiments, separate tables store the responses for each experiment. The name of the response table for a particular experiment is identified by the response_table field in the experiment table. Although it would have been possible to store responses for every experiment in a single table, the response tables are separated in order to add a measure of data protection and ease of use. The structure of a response table is shown in Figure 10.

The response_id field is the primary key of the table. It is simply an integer that increments with each successive record to uniquely identify a record in the table. The response_order field indicates the relative order of the response for the session. All responses submitted on a particular form have the same response order, since all questions of a form are submitted to the database at the same time, even though the entries for each question are made separately. The response_order is particularly useful if a form is used in an experiment with a loop, since the field value increments with each form submission. The date_time field indicates the date and time the submission was made.

The response itself is stored in either the response_text or response_enum field. If the data format associated with the question is a predefined type, such as int16, double, or varchar, the response is stored in the response_text field, while if it is of an enumerated type, it is stored in the re-



Please enter the following information:
Please do not click the "back" or "refresh" browser buttons as this will re-submit your responses.
Click the "Next" button at the bottom of the form to go to the next page.

1) Please indicate which hand you prefer to use for each of the following activities. If you use your left and right hands interchangeably, select "ambidextrous."

writing
Left

drawing a picture
Left

throwing a ball
Ambidextrous

brushing your teeth
Right

cutting with scissors
Left

Figure 8. Multiple part question with a "null" subquestion data type.

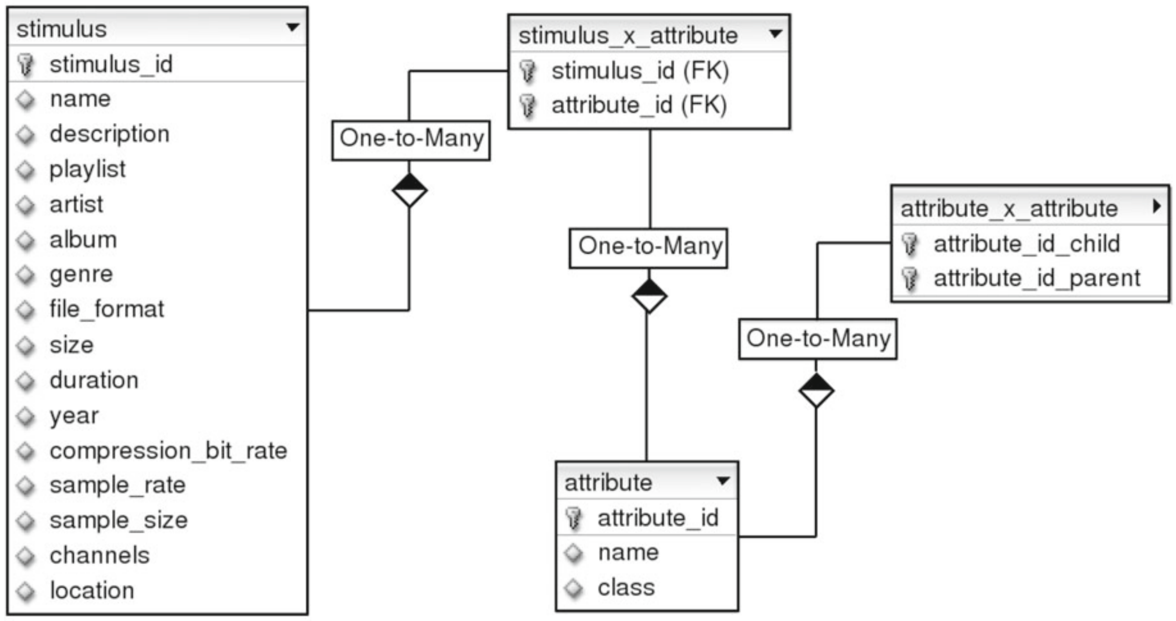


Figure 9. Stimulus and attribute schema. See Figure 7 for a description of the items.

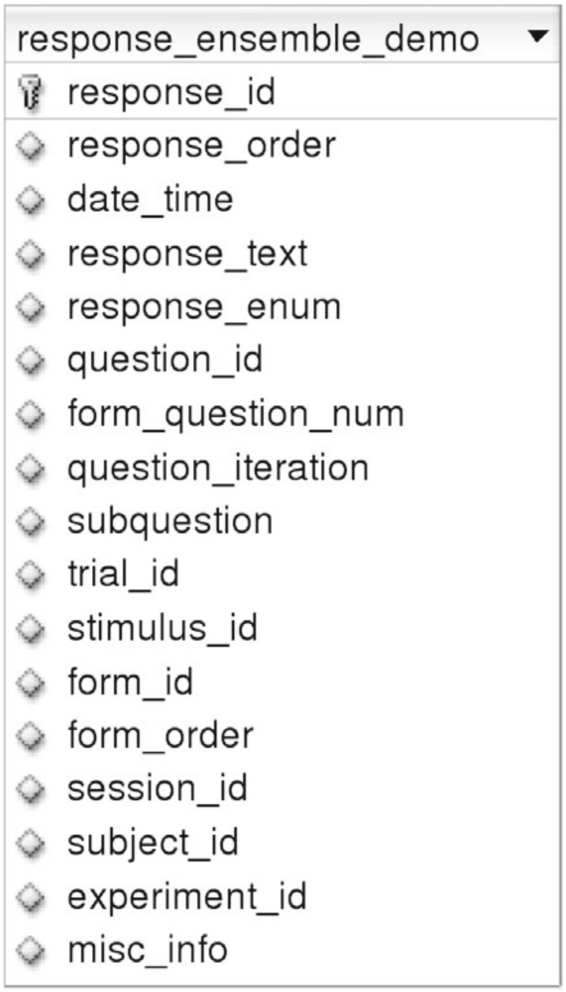
sponse_enum field. All responses stored in the response_text field are stored as plain text in the MySQL database. A wide range of character sets may be assigned to text fields in MySQL for support of various languages. Text fields in MySQL are variable in size, consuming only two bytes more than the amount of storage space required for the submitted character string. The same limit of 65,536 characters for the "text" response type applies to text fields in the database. A text field fulfills the need to store values for all of the predefined types in a single field. We found it more desirable to have the ability to store all responses in as few fields as possible, rather than create a response field for each data type. This reduced the need to alter MATLAB analysis code when retrieving different types of responses. Some storage efficiency is sacrificed by storing multiple data types as text. Data constraints for predefined types are enforced by PHP scripts in the QPI and are checked by the PHP scripts before submission to the MySQL database. A response that violates a type constraint results in an error message that requests the participant to re-enter the response.

Enumerated types are stored in the response_enum field as 64-bit integers. A bit mask indicating TRUE or FALSE values for the enumerated elements is converted to an integer for this purpose. While the use of a 64-bit integer imposes a limit of 64 elements for each enumerated set, we view this limit as acceptable, given that questions for which a participant must chose from among 64 possible response items are unlikely to arise. The use of a bit mask is the most efficient method of storing an enumerated response. Since most of our responses are enumerated, this compensates for the lack of efficiency in storing responses of one of the predefined types.

Records in the response table are linked to the question table by the question_id field. The fields form_question_num and question_iteration link to the form_x_question table and specify the order of the question on a particular form and the number of possible answers to the question (e.g., for entering each language that the participant fluently speaks when asked to list them). The subquestion field is a foreign key to the question_x_data_format table and identifies the part of a multititle question or a multiquestion. The trial_id and stimulus_id fields identify the trial and stimulus associated with the response, if applicable.

The form_id and form_order fields are references to the form table, and identify the exact form and the order of occurrence of that form in the experiment. Each response identifies the session, participant, and experiment through the session_id, subject_id, and experiment_id fields. The misc_info field can be used for a number of purposes. Currently, we use this field to identify responses that were copied from previous sessions when the participant revisits a "visit once" form.

The schema of Ensemble provides a high degree of flexibility for implementation of questions of many different types. The ability to reuse questions enables one to standardize them, i.e., use them across many surveys and experiments simultaneously without concern of discrepancy of wording or response format. The recycling of forms among experiments likewise standardizes them and reduces data redundancy. On the other hand, the high degree of flexibility of the schema also introduces complexity to the system. However, this issue is overcome in the analysis phase by the development of generic MATLAB scripts that can be reused among experiments with little or no modification. Complexity is not an issue in the development phase, as the submission and organization of data is mostly managed by the QEI.



response_ensemble_demo
response_id
response_order
date_time
response_text
response_enum
question_id
form_question_num
question_iteration
subquestion
trial_id
stimulus_id
form_id
form_order
session_id
subject_id
experiment_id
misc_info

Figure 10. Response table structure. See Figure 7 for a description of the items.

MATLAB Workspace

MATLAB is a commercial application that provides a scripting language and a wide range of utilities for numerical analysis. Since a satisfactory description is beyond the scope of this paper, those unfamiliar with MATLAB are encouraged to read a description at www.mathworks.com/products/matlab. The MATLAB workspace allows researchers to write custom functions for analysis of experiment data. A utility called MySQL Database Connector (Almgren, 2005) retrieves data from the MySQL database. A number of general-purpose MATLAB scripts use this utility to ease the retrieval of data for analysis.

In addition, we provide a utility for exporting responses to a comma delimited text file for those who do not wish to use MATLAB. The researcher specifies the experiments, sessions, participants, and tickets to query. The utility resolves enumerated responses, which are stored in the database as integers, to the matched enumerated value or values selected by the researcher. Data obtained from the response table are joined to the question, question_x_data_format, and data_format tables so that the question text and data types are also exported with the responses. The exported text file can then be directly imported to an alternate analysis application such as SPSS, SAS, or Statistica.

MatRPC

MatRPC, a lightweight threaded application written in C, was developed by our lab for use with Ensemble. The application runs as a daemon, or background process, on our server. MatRPC sends and receives messages via TCP/IP network sockets. A TCP/IP socket is a mechanism by which messages are passed between two applications on the same computer or two different computers on a network. MatRPC only supports messages which are passed as strings. Each string sent to MatRPC includes a session identifier and a MATLAB function along with its function arguments. The only MATLAB functions currently supported are those which return a single string. When a message containing a new session identifier is sent to MatRPC, a MATLAB engine is created using the routines provided in the MATLAB API. MATLAB engines operate in a similar fashion to MATLAB workspaces insofar that each engine has its own variable space. However, no user console is provided and the engines are controlled by the daemon. When a new call to MatRPC repeats a session identifier, the same MATLAB engine is called. The session identifier sent to MatRPC is identical to the session ID field used in the MySQL database, thereby ensuring that each is unique and providing a straightforward method to match a QPI session to a MATLAB engine instance. The MATLAB function is called through the engine and the result is sent back to the QPI, which interprets the returned string to play a stimulus, present a break, or skip a series of forms.

During presentation, a QPI form handler associated with one of the forms in the loop calls the MATLAB function through MatRPC and a stimulus ID is returned as a string. The form handler queries the database for the path and filename of the stimulus based on the stimulus ID, then passes this information to the Flash player to present the stimulus before presenting the questions associated with the form. After a specified number of stimuli have been presented, the MATLAB function returns the string “break,” which signals the form handler to present an instruction to the participant to take a break if desired. The form handler presents the participant with a button to continue with the experiment when ready. The MATLAB function continues to return stimulus IDs and “break” strings until all of the stimuli have been presented for the session. Once finished with stimuli, the MATLAB function returns the string “end,” signaling the form handler to skip a number of forms defined in the form handler to break out of the loop.

MatRPC was developed so that researchers would not have to write real-time analysis scripts in PHP. However, MATLAB and MatRPC are not required in order to use most of the features in the rest of Ensemble, and most of the more sophisticated logic for which MATLAB is used could be written in PHP or an alternative scripting language.

Flash

Flash is an application that allows developers to provide video, audio, and user interactivity to Web pages. Most Web browsers are bundled with a Flash player plug-in for presenting Flash movies. The player, if not already installed on a workstation, can be downloaded and installed free of charge. Two Flash movies are currently bundled with Ensemble. Both movies play mp3 audio files stored in the server's file system. The first movie simply plays a single audio stimulus without providing any graphics or controls to the participant. A second movie provides a play button and can either play a single audio stimulus or two audio stimuli in succession. Many Flash movies that employ audio and/or video can be easily incorporated into the Ensemble framework. The only requirement for the flash movie is that it must accept a variable that indicates the Web page to load after it finishes playing. The QPI passes the name of a script to the Flash player that redirects to the current page of the QPI session. The Flash player loads this page when it finishes playing, allowing the QPI session to continue.

phpMyAdmin

PhpMyAdmin is a Web-based administration utility written in PHP for managing a MySQL server. The application provides a graphical interface for performing many administration tasks which are possible but cumbersome with MySQL text based utilities. PhpMyAdmin is used in our lab for designing and maintaining the Ensemble database. Additionally, it is sometimes used by researchers in the lab to edit question records or manage stimulus records.

DATA SECURITY

MySQL has its own user privilege system for access control to the database. In addition to specifying user privileges to databases and tables, the system can be used to restrict access from specific TCP/IP domains (i.e., computer networks) or specific computers. This privilege system is independent of the users and privileges employed for access to the operating system of the server. Ensemble has two MySQL users for accessing the database: a "participant" user is necessary for enabling the participant to submit responses to the database through the QPI, and an "experimenter" user is granted much wider access to the database for managing experiments, forms, and questions through the QEI. Even though a "participant" username and password exist, the participant has no notion of them when using the QPI. The username and password are used in the background by the QPI scripts to provide access to the MySQL database, and are not sent in any fashion to the Web browser that the participant is using. In the case that the PHP scripts and MySQL server are on the same server (the recommended configuration), the "participant" username and password are passed without being sent over a network, providing additional security. The "participant" username and password are stored in a secure directory outside of the Web directory tree (i.e., not publishable via the Web).

Only the researcher has a notion of a username and password. He or she is prompted for an "experimenter" username and password immediately upon launching the QEI. The researcher's username and password is only transmitted over the network once, when the QEI is first initiated. Afterward, the username and password are temporarily stored in two PHP variables on the server for the duration of the session.

PhpMyAdmin supplies a user-friendly interface for managing MySQL users. Privileges can be assigned at the database or table level. Using this interface, one could restrict researcher privileges over the subject table to protect participant anonymity. A second researcher account can easily be created with access to the subject table, thereby providing a multitiered administration structure for Ensemble.

MySQL provides the ability to restrict access to users from specific TCP/IP domains, or computer networks, and specific computers. The "participant" user does not need access from any computer except the server where the PHP scripts are installed, since the login and password are exclusively used in the background. The "experimenter" user can likewise be restricted to the server, since the username and password are used by the scripts in the background once submitted via the Web browser. However, if MATLAB analysis scripts are employed, the "experimenter" user must be granted access from the computers where the scripts are launched. Even though it would be very difficult for a computer hacker to access any data without acquiring a username and password, the ability to restrict access to specific networks and computers provides additional security.

One can also restrict access to the QEI and phpMyAdmin Web pages regardless of user restrictions. All of the QEI scripts are located in an "admin" subdirectory of the root Ensemble directory. Since phpMyAdmin is a utility developed by another group, it is installed in its own directory as well. The Apache Web server can therefore be easily configured to restrict access to these directories except from specific computers and networks. In the event that access is attempted from nonauthorized computers or networks, an error page would result instead of a login screen. Apache may also be configured to require only secure connections to Ensemble (i.e., using the same encryption technology employed by online shopping sites). Many Web servers besides Apache also provide this capability. If this is enabled, the "experimenter" username and password and any data submitted through the QEI, QPI, and phpMyAdmin are securely transmitted over the network.

We recommend that Ensemble be restricted to secure Web connections. Additionally, the Web server should only grant administrative access to necessary locations (e.g., the research lab). The "experimenter" user can additionally be restricted to these locations to prevent access from any other MySQL client.

The configuration of usernames and passwords for MySQL is handled through the installation process of Ensemble. We have also provided a description of our Apache configuration with the Ensemble distribution so that users could duplicate it on their own installations.

DISCUSSION

Although the development of Ensemble necessitated a substantial time investment (the project required approximately 1.5 years to bring to its current stage), it is now a system which saves many hours in experiment administration and significantly improves the organization of experiments. Ensemble is built on environments which are used widely and are largely platform-independent. PHP scripts, MATLAB scripts, Flash movies, and MySQL schema are easily ported between the three most popular platforms: Linux, Mac OS X, and Windows with little or no adjustment. Further, upgrades to these platforms are usually backward compatible to a high degree and require few or no changes to existing code. We therefore expect maintenance of the current codebase of Ensemble to be minimal, and any future work to be focused primarily on extending its functionality. One extension we plan to add is improved ability to view different categories of questions and forms. This will prove useful when a large number of questions and forms are added to Ensemble.

Placement of the MySQL database at the center of the framework eases the submission and retrieval of data when transitioning from one modality of working on the experiment to the next. The QEI, QPI, MATLAB, and phpMyAdmin modules reference the same tables and records, and each uses a MySQL client that communicates with the database via SQL statements. Therefore, the researcher may use the same SQL queries to obtain data such as responses or question text in phpMyAdmin and the MATLAB workspace.

Further, the integration of the data acquisition and analysis paths significantly frees the researcher from the burden of exporting and importing data from one format to another. One does not need to worry about issues of data file organization and proper importation of records. Automatic organization of data allows the researcher to focus on tasks more relevant to his or her research.

The schema of Ensemble may at first glance appear to be complicated and therefore require a substantial investment of time for familiarization. However, the researcher often does not need to concern himself or herself with the underlying structure of the data when creating and organizing the experiment. This is almost entirely handled by the QEI. Some attention to the schema is necessary during the analysis phase. However, the creation of generic MATLAB scripts for retrieving data largely frees one from concern with the data organization. We have introduced several undergraduate and graduate student researchers to Ensemble and they have learned to use it for experiment creation and analysis in a modest amount of time, ranging from half a day to 2 days.

While Ensemble provides a multitude of utilities, its function set is by no means exhaustive. It is not capable of recording response times, for instance, a function which several other experiment management programs such as Presentation or Paradigm (López-Bascuas, Carrero, & Serradilla, 1999) offer. Recording of response times might be possible in Ensemble through custom Flash movies. This hasn't been tested, however, so we cannot attest to the accuracy of this method. Paradigm also provides the capability of specifying fixed intertrial intervals. While this is not a feature of any of the current form handlers in Ensemble, it might be possible to provide this function through a new form handler or with a Flash movie. Other tools, such as Generic HTML Form Processor (Göritz & Birnbaum, 2005) seek to provide a general purpose interface. In Ensemble, we attempted to create a set of utilities to ease the management and analysis of simple surveys, with extensions for stimuli presentation and integration with MATLAB.

Ensemble is open-source and we invite other groups to use it, add new functions and modules, and publish their revisions. It is available for download along with installation instructions at atonal.ucdavis.edu/ensemble/installation.

AUTHOR NOTE

This work was supported in part by NIH Grant R03 DC05146-01 to P.J. We thank Jason Golubock for writing MatRPC. Correspondence regarding this article should be addressed to P. Janata, Center for Mind and Brain, University of California, 267 Cousteau Place, Davis, CA 95618 (e-mail: pjanata@ucdavis.edu).

REFERENCES

Almgren, R. (2005). MySQL Database Connector [Computer software]. Retrieved February 13, 2006, from www.mathworks.com/matlabcentral/fileexchange.

Göritz, A. S., & Birnbaum, M. H. (2005). Generic HTML Form Processor: A versatile PHP script to save Web-collected data into a MySQL database. *Behavior Research Methods*, **37**, 703-710.

López-Bascuas, L. E., Carrero, C., & Serradilla, F. (1999). A software tool for auditory and speech perception experimentation. *Behavior Research Methods*, **31**, 334-340.

phpMyAdmin Project (2005). phpMyAdmin (Version 2.6.3-pl1) [Computer software]. Retrieved July 15, 2005, from www.phpmyadmin.net.

W3C Validator Team (2006). *W3C markup validation service*. Retrieved July 2, 2006, from validator.w3.org.

Zinner, M. G. (2003). DBDesigner (Version 4.0.5.4 Beta) [Computer Software]. Pressbaum, Austria: Fabulous Force Database Tools.

NOTES

1. One is limited only by the maximum MySQL table size, which depends on the operating system and table file format used. The limit is typically large, ranging between 2 GB and 64 TB.

2. Approximately 1.5 MB is used by PHP scripts and MatRPC. Most disk space consideration is based on the storage of stimuli and MySQL records, which depend on specific research needs.

(*Continued on next page*)

APPENDIX
System Recommendations

Ensemble, consisting primarily of PHP scripts, MATLAB scripts, and MySQL database schema, can be implemented on a number of computer platforms which support these applications. PHP, MySQL, and MATLAB can be installed on the three most popular operating systems: Linux, Mac OS X, and Windows. The QPI, QEI, and phpMyAdmin were written in platform independent PHP code, so porting them to a new operating system should be straightforward. The MySQL schema is similarly platform independent, and uploading the MySQL tables should be easy once the MySQL server is installed. MATLAB code is likewise portable between operating systems. The Flash program for developing Flash movies is only available for Windows or Macintosh systems, although a player for presenting movies is available for each of the three mentioned platforms. The component with the narrowest requirements is MatRPC, which runs on the Linux operating system. MatRPC is a multithreaded application programmed in C and uses standard C libraries and POSIX threads, so it may likely be ported to other operating systems with little effort. It need not run on the same computer as the other components of Ensemble, although it must be installed on a computer supplied with a MATLAB installation.

The QEI and QPI should work well with a variety of browsers, although so far we have only tested a few of them. We have extensively used Firefox on Linux, Mac OS X, and Windows for the QEI, and used the Mac OS X and Windows versions of this browser for the QPI without any issues. The Flash player on Linux isn't well supported and for this reason we do not recommend using the QPI on Linux, at least when the Flash player is required. Since Firefox is a basic Mozilla-based browser, incorporating other Mozilla-based browsers such as Netscape should not be problematic. We have also used Safari on Mac OS X extensively for both the QEI and QPI without any browser issues. Although we are not aware of any incompatibilities with Internet Explorer, we haven't extensively tested the interfaces with this browser. PhpMyAdmin has been validated for the XHTML 1.0 standards documented by the W3C Markup Validation Service (The W3C Validator Team, 2006), so the utility should generally be compatible with the latest browsers on all operating systems. Our goal is general compatibility with all current browsers so we will address issues with other browsers as they arise.

The server which hosts Ensemble in our lab has the following hardware and software components. These are by no means minimum requirements and are provided for informational purposes.

Hardware

CPU: Intel Pentium III 600 MHz (dual processor)
Disk Space: 66 GB[2]

Software

Operating System: Redhat Linux Enterprise WS, version 3.0
Web Server: Apache, version 2.0.46
PHP: version 4.3.2
MySQL Server: version 4.0.26
phpMyAdmin: version 2.6.3-pl1
MatRPC: version 1.0 (developed by our lab and available for download)
MATLAB:versions 6.5.1.199709 (R13) Service Pack 1 and 7.1.0.183 (R14) Service Pack 3
Flash: version 2004 MX Professional

(Manuscript received March 10, 2006;
revision accepted for publication August 30, 2006.)