The Capture and Use of Detailed Process Information in the Dialogix System for Structured Web-Based Interactions

Thomas M. White, MD, MS[1,2] and Michael J. Hauan, MD, MPH, MTS, MA[1,3],
[1]Columbia University Department of Medical Informatics, [2]New York State Psychiatric Institute,
[3]Columbia University Division of Child & Adolescent Psychiatry, New York, NY

Information gathering tools, such as questionnaires, surveys, and structured interviews, are ubiquitously used in evaluating patients and systems. Despite their common use, there is a desperate need for better questionnaires in medical research[1] and epidemiology[2], and an infrastructure that lets them be publicly scrutinized.[3]. Unfortunately, there has been no common platform that supports the deployment of arbitrary information gathering tools.

Some psychiatric diagnostic interviews and epidemiological trials require sophisticated structured interviews containing complex branching logic, dynamic phrase composition, and multiple languages. The Dialogix system was developed to meet this need and facilitate the rapid definition and web-based deployment of structured human-computer interactions.

This paper describes the content and process-related information captured by Dialogix, and how that information has been used in the development and deployment of two large epidemiological studies.

INTRODUCTION AND BACKGROUND

Customizing the presentation and collection of information is a common challenge for many fields. Public health and marketing research include efforts to tailor information for subjects by changing the themes, wording, and presentation of the messages based upon data gathered about the subjects. Similarly, complex questionnaires and structured diagnostic interviews use internal logic to skip past irrelevant questions; and some even tailor the content of the instructions and questions themselves. Likewise, computer assisted instruction and testing systems might benefit from adjusting the information presented and/or questions asked based upon prior responses.

From a researcher's perspective, these systems have quite different goals, challenges, and back-end processing needs. From a user's perspective, however, these systems are all variants on the theme of reading an interactive electronic book (i-Book). Users can see only one page (screen) of information and/or questions at a time, along with buttons allowing them to navigate forward or backwards through the i-Book one page at a time. Answers to the current questions determine which page the user will see next, thus allowing irrelevant material to be skipped without burdening the user with the complexity of the back-end processing. Unlike a typical web-browsing experience, users can not search among or navigate to arbitrary pages. Instead, i-Books have a distinct start, middle, and end; and only the current page can be bookmarked.

The Dialogix system is a collection of Java-based servlets designed to facilitate the creation and deployment of such i-Books. Internally, Dialogix uses the page as the primary transactional unit, with each page containing any number of tailored messages or questions. Standard web-browsers, stripped of their own navigation bars, are used to display the i-Book pages and their navigation buttons. Between pages, the back-end system validates and logs the answers, and uses this new information to determine the content of the next page. When users skip required questions or provide answers that fail internal validity checks, the system re-sends the same page, with helpful error messages. If no input errors occur, the system sends a page composed of the next set of logically reasonable messages and questions. Users can also navigate backwards to review prior pages, optionally changing answers and thereby potentially altering the content of future pages.

One of the main goals for Dialogix was to enable researchers and educators to rapidly create, polish, and deploy complex instruments without needing programmers, data managers, or web-masters. We have achieved some success, as evidenced by the wide variety of tools that have been created using Dialogix, including surveys, decision trees, clinical guidelines, and guided tutorials.

METHODS

The development and conceptual schema underlying the Dialogix platform are described elsewhere.[4] Briefly, Dialogix was developed iteratively, with

1067—5027/01/$5.00 © 2001 AMIA, Inc.

ongoing usability and needs analysis. We elected to use transaction based logging to allow path tracing and state reconstruction. JavaScript was used to collect client-side usability information.

Dialogix Data and Event Files

Table 1: Data File (one row per node)	
Column	*Description*
varName	the unique variable name
stepNum	index of this step within the schedule
groupNum	index of this group of questions within the schedule
langNum	the language in which question was asked
question	the exact text of the question, complete with micro-tailoring
answer	the answer the subject gave
comment	an optional comment
timeStamp	time, in milliseconds, when the group of questions was answered
when	number of seconds since start of the interview
duration	number of seconds spent on the group of questions

Table 2: Raw Event File (one row per event)	
Column	*Description*
dispCnt	count of the number of screens of information the subject has seen
varName	the unique variable name
actionType	radio, select, text, textarea, submit
eventType	load, focus, click, keypress, change, blur
timeStamp	absolute start of this event, in milliseconds
duration	duration of this event, in milliseconds
value1	additional event information - the key pressed, or the select value
value2	human-readable representation of the interaction - the text of the select box

During the course of the structured interaction, Dialogix captures and logs both content and process information. The content data is stored in the data file (Table 1). As would be expected, Dialogix stores every answer entered. Unlike traditional systems, however, Dialogix uses a log-file format, thus storing the complete history of changes to the answers. Moreover, since Dialogix supports micro-tailoring of the instructions, questions, and answer-options, the data file also stores the exact wording of the messages and questions the subject saw, as well as the language in which it was asked. Timing information is stored, including the amount of time spent on that page and the cumulative time since the start of the interview. Finally, the data file stores the position of the node in the interaction file, and which potential page it is in. These positional variables allow assessment of the trajectory through the interview.

The raw event file (Table 2) stores the type and timestamp of both server and client-side events. It also stores a running count of the number of page displays the user has seen. Like the Data file, it is transaction based, with all new events appended to the end of the file. Tables 3 and 4 show the types of data that are calculated from a combination of Tables 1 and 2.

Table 3: Events Per Page (one row per page)	
Column	*Description*
dispCnt	count of the number of screens of information the subject has seen
groupNum	index of this group of questions within the schedule
loadMs	load time: milliseconds between when the browser receives the input and when the screen is first available for view or input
dispMs	display time: milliseconds of client-side display and processing, not counting loadMs
turnMs	turnaround time: milliseconds between when server sends questions to the client and when it receives the response to that set of questions
ntwkMs	total network time in milliseconds = (turnMs - dispMs)
loadDate-Time	timestamp of when the screen was first loaded - in year/month/day hour:min:sec.ms format
loadDiff	hour:min:sec.ms difference between two consecutive loadMs

The server side events include the time that a request was received, and the time that the next page was sent back to the client. Interval data is dynamically calculated from this, including the amount of time required for the server to process the request and the turnaround time: the amount of time the client spent interacting with the page.

Table 4: Events Per Question (one row per event)	
Column	*Description*
dispCnt	count of the number of screens of information the subject has seen
stepNum	index of this step within the schedule
groupNum	index of this group of questions within the schedule
count	the index of this question on the screen: 1-num
type	the type of question (*actionType*)
totalTime	cumulative time spent focused on the question (blur-focus)
inputTime	cumulative time spent entering and changing the answer to this question (blur-<first non-focus event>)
answered	count of times the question was answered on this screen
skipped	number of times the question was skipped on this screen

On the client side, all JavaScript events are captured and logged, except for mouse movements. These include focus events (when an input item is first ready to receive data - e.g. when the cursor starts blinking in a text box, or when a select box becomes highlighted), blur events (when the focus leaves an item). From these, the total amount of time spent on a question can be calculated, which may include both the amount of time spent reading the question and also the time spent entering the answer. Input events are also captured, including key presses and mouse clicks. From these, it is possible to calculate the amount of time spent entering each answer. Finally, the load event is captured. This makes it possible to determine how long it took the client browser to display the page after receiving it from the server.

RESULTS

The ability to collect this process data has allowed us to address several concrete issues in the epidemiological studies using Dialogix.

Children in the Community (CIC) Study

The CIC study[5] has been tracking 800 subjects over the last 25 years. A major goal is to assess how environmental and quality of life issues correlate with and predict psychopathology. Dialogix was chosen to computerize the current (sixth) wave of data collection on this cohort. The components of the study are listed in Table 5, and are all based on existing instruments. In all, there are 968 questions, of which 457 are required, and 511 can potentially be skipped (e.g. if the subject is not employed, or is mentally healthy).

Table 5: CIC Study[5]		
Section	**# questions**	**# skippable**
Quality of Life		
Residence	55	0
Work	43	43
Education	36	36
Homemaker	16	16
Financial	35	25
Social	93	66
Health	54	29
Self Report Traits		
NIDA	105	0
CIC	100	0
SCID I Screen	119	100
SCID II Screen[6]	116	0
Clinical Assessment		
SCID I Followup[7]	92	92
SCID II Followup[6]	104	104
Totals	**968**	**511**

Since response quality decreases as subjects lose interest, and possibly with instrument length[1] one goal of computerization was to keep the as short as possible. Specifically, the designers wanted to limit the average interview to 2.5 hours. Since prior to the computerization they had no formal way of assessing how long it took to complete each module, or which questions were consuming the most time, they group decided to ask the minimum number of questions necessary and sufficient to rule-in or rule-out the diagnosis of each DSM-IV[8] Axis II personality disorder. This was unfortunate, since it would be clinically useful to know whether, for example, a subject had the 5 symptoms necessary for a diagnosis of Borderline Personality Disorder, or all 9 possible symptoms. Moreover, several Axis I disorders were dropped from consideration for fear that assessing them would be too time consuming.

Using the Dialogix process data from the first 20 pilot interviews, it was possible to assess the average time per question, clustered by module. It was found that the average interview lasted 1 hour and 45 minutes, with a standard deviation of 25 minutes. The Clinical Assessment section was taking a median of 10 minutes (range 7.3 to 19.7, with one outlier taking 71 minutes). Most importantly, it was determined that questions in the Clinical Assessment section took a median of 34 seconds to answer (range 28-41 with an outlier at 88 seconds). Thus, since asking all relevant follow-up questions for Axis II entails 4-5 additional questions per diagnosis, this translates to about 2 minutes extra per diagnosis. This information is being used to redesign the study before final deployment. Given that the interviews were already shorter than expected (based upon estimates of how long it took for the paper-based interviews), and given how little extra time it would take to collect complete symptom information instead of just dichotomous diagnosis information, many of the questions that were initially cut may be incorporated into the final version.

Boricua Youth Study (BYS)

The BYS Study[9] is a new longitudinal cohort study of 2800 Puerto Rican Youths, half in New York City, and half in mainland Puerto Rico. The goal is to assess what epidemiological factors contribute to prior observations that mainland Puerto Rican youths meet criteria for Antisocial Personality Disorder as adults without having met criteria for Conduct Disorder as children. This conflicts with current theory about the development of violence. In contrast to the CIC study, the BYS study is mostly comprised of novel questionnaires, plus the C-DISC[10] - a separately computerized instrument. Both the parent and child are interviewed, with an expected average interview time around four hours. Table 6 contrasts some characteristics of these studies.

Several concerns were raised during the implementation of the BYS that can be addressed using the process data Dialogix collects.

Table 6: Studies			
	BYS Adult	BYS Youth	CIC
nodes	1534	1313	1783
questions	1134	1207	968
always asked	386	229	354
dependent	748	978	511
open-ended	97	60	0
instructions	75	73	441
evaluations	206	106	281

First, there was considerable debate about whether subjects should be allowed to go back and change answers, especially if it meant going back more than a dozen questions. Although we never fully understood the rationale for this concern, we realized that we could address it using Dialogix. Foremost, since Dialogix stores a complete log of answers, and any changes that occur, there is no risk of data loss. Moreover, Dialogix process data can be used to evaluate how frequently subjects changed questions, and whether there are any cross-subject patterns to this behavior that might suggest systemic problems in the study. One convenient visualization might be to plot a histogram of *stepNum* vs. the number of times it was present in the log file. Such a graph would show which questions were revisited. The same data could be used to assess how often this resulted in changed answers or different down-stream branching; as well as to assess how much time was consumed in the process.

Another major concern is quality assurance. Since the interviews are long, and the subjects and interviewers are only modestly compensated for their time, there is the risk that subjects and interviewers might record random answers as they get tired or lose interest. All interviews are tape-recorded. The first two interviews done by each interviewer are fully reviewed. After that, a subset of tapes is spot-checked. Dialogix may facilitate this process. Since time-per-question is recorded, it will be possible to detect disturbing trends, such as cases where questions are answered more quickly than reasonable (e.g. looking for outliers when assessing the median response time for questions).

Another quality assurance concern is related to the coding of open-ended questions. The BYS contains 157 questions of the type "other, please specify". Few of those are likely to be asked. However, it is imperative to evaluate and code their contents. Dialogix log files are being used to validate that these questions are being reviewed and coded. Specifically, post-processing scripts are run on the data files to determine which questions need to be coded, and to set a flag to turn on reviewer mode. Then, the reviewer uses the *jump-to* button to get to the questions that need coding, letting them review the answers within the context of the surrounding questions. Since they are in reviewer mode, the open-ended questions are followed by questions asking them to store coded values. Thus, Dialogix is able to log the amount of time they spend reviewing and coding the questions. Questions that are coded too quickly given the length or complexity of the answer can be detected and flagged for quality assurance review.

DISCUSSION

The data collected via Dialogix has been used to inform the development of two epidemiological studies. This same data might also be of benefit in other domains, including usability, education, network administration, and expert review.

The process data collected via Dialogix is comparable to that collected in expensive usability studies[11]. Like usability labs, Dialogix captures a record of user events in such a way that their navigation through the system can be reproduced and evaluated. Usability studies often go several steps further, audio-taping and videotaping the human-computer interaction. They may also use pluggable User Interface Management Systems (UIMS) to collect frequent screen-shots to monitor all mouse and keyboard events. In formal usability studies, the most valuable insights tend to come from observing the subjects as they struggle through and talk about the task. Although copious keyboard and mouse information is gathered, it is rarely utilized.

In some ways, Dialogix could be considered a pluggable, discount usability-engineering platform. Any structured interaction implemented using the Dialogix system can benefit from the data it captures.

There are several obvious usability studies that could be performed using Dialogix. Similar questions can be assessed using different input styles, such as radio buttons, list boxes, and combo boxes. The inputTime variable can be used in conjunction with an integrated satisfaction questionnaire to assess which input style is best under different circumstances.

These process data also have potential benefit for web-based education. Many of us have had the experience of seeing a question late in an exam that makes us want to go back and change the answer of a previous question. A common concern of is the prevention of this behavior in web-based testing systems.[12], especially those that provide the answer key at the end. With Dialogix, however, it is possible to track the actual navigation of users through a testing module. Not only does this support the detection of changed answers, in cases where answer keys are given, but it also allows for assessments of unanticipated interdependencies among questions (e.g. those that give away the answer to prior questions).

These data can also inform infrastructure decisions. Since server processing times and network times are captured, it is possible to detect sources of bottleneck for centralized administration of questionnaires. For example, if the server processing time is long relative to the network delay times, a multi-server, load-balanced design might be preferable. Likewise, network delay times can inform wiring and router decisions. Finally, the loadTime variable shows the effects of different browsers and client platforms. These data can be used to inform decisions about minimum necessary hardware and software requirements on the client side.

The Dialogix process data might also help in the design of questionnaires themselves. The quality of the questions themselves is seldom assessed, often due to time and cost constraints. Another factor is that there is often considerable time between making a text change and having it available in a testable prototype. A common method for assessing the quality of questionnaires, guidelines, and other health communication is the Delphi Technique. [13]. Dialogix's ability to rapidly design and deploy questionnaires might make it a valuable tool in such collaborative work.

CONCLUSION

We believe that Dialogix is a unique example of how a rich set of process information can be collected during the course of a web-based interaction. These data have been valuable in the design, deployment, and quality assurance of two large epidemiological trials. These data, combined with Dialogix's support for rapid, iterative development, may facilitate the rapid design, testing, and refinement of a wide variety of instruments.

REFERENCES

1. Eaden J, Mayberry MK, Mayberry JF. Questionnaires: the use and abuse of social survey methods in medical research. Postgrad Med J 1999;75(885):397-400.
2. Olsen J. Epidemiology deserves better questionnaires. IEA European Questionnaire Group. International Epidemiological Association. Int J Epidemiol 1998;27(6):935.
3. Wilcox AJ. The quest for better questionnaires. Am J Epidemiol 1999;150(12):1261-2.
4. Hauan MJ, White TM, Johnson SB. Dialogix: a system to computerize structured instruments. in preparation.
5. Cohen PR. Children in the Community (CIC) Study. In: Accounting for Change in Psychopathology - ages 17-25: NIMH Grant MH54161-01; 2000.
6. First MB. User's guide for the structured clinical interview for DSM-IV axis II personality disorders : SCID-II. Washington, DC: American Psychiatric Press Inc.; 1997.
7. Spitzer RL, National Institute of Mental Health (U.S.). User's guide for the Structured clinical interview for DSM-III-R : SCID. Washington, DC: American Psychiatric Press; 1990.
8. American Psychiatric Association., American Psychiatric Association. Task Force on DSM-IV. Diagnostic and statistical manual of mental disorders : DSM-IV. 4th ed. Washington, DC: American Psychiatric Association; 1994.
9. Bird HR, Canino G. Boricua Youth Study (BYS). In: Antisocial Behaviors in US and Island Puerto Rican Youth (5/1/98 - 2/8/03): NIMH Grant MH56401; 2000.
10. Jensen P, Roper M, Fisher P, Piacentini J, Canino G, Richters J, et al. Test-retest reliability of the Diagnostic Interview Schedule for Children (DISC 2.1). Parent, child, and combined algorithms [published erratum appears in Arch Gen Psychiatry 1995 Apr;52(4):312]. Arch Gen Psychiatry 1995;52(1):61-71.
11. Nielsen J. Usability engineering. Boston: Academic Press; 1993.
12. Schleyer TK, Forrest JL. Methods for the design and administration of web-based surveys. J Am Med Inform Assoc 2000;7(4):416-25.
13. Snyder-Halpern R, Thompson CB, Schaffer J. Comparison of mailed vs. Internet applications of the Delphi technique in clinical informatics research. Proc AMIA Symp 2000:809-13.