turning knowledge into practice

The Joy of Developing a Web-based Survey System!

Stuart Allen, Brandon Peele, Chris Rasmussen,
Sridevi Sattaluri, R. Suresh, Emily Warmoth

FedCASIC 2009

March 18,2009

RTI International is a trade name of Research Triangle Institute

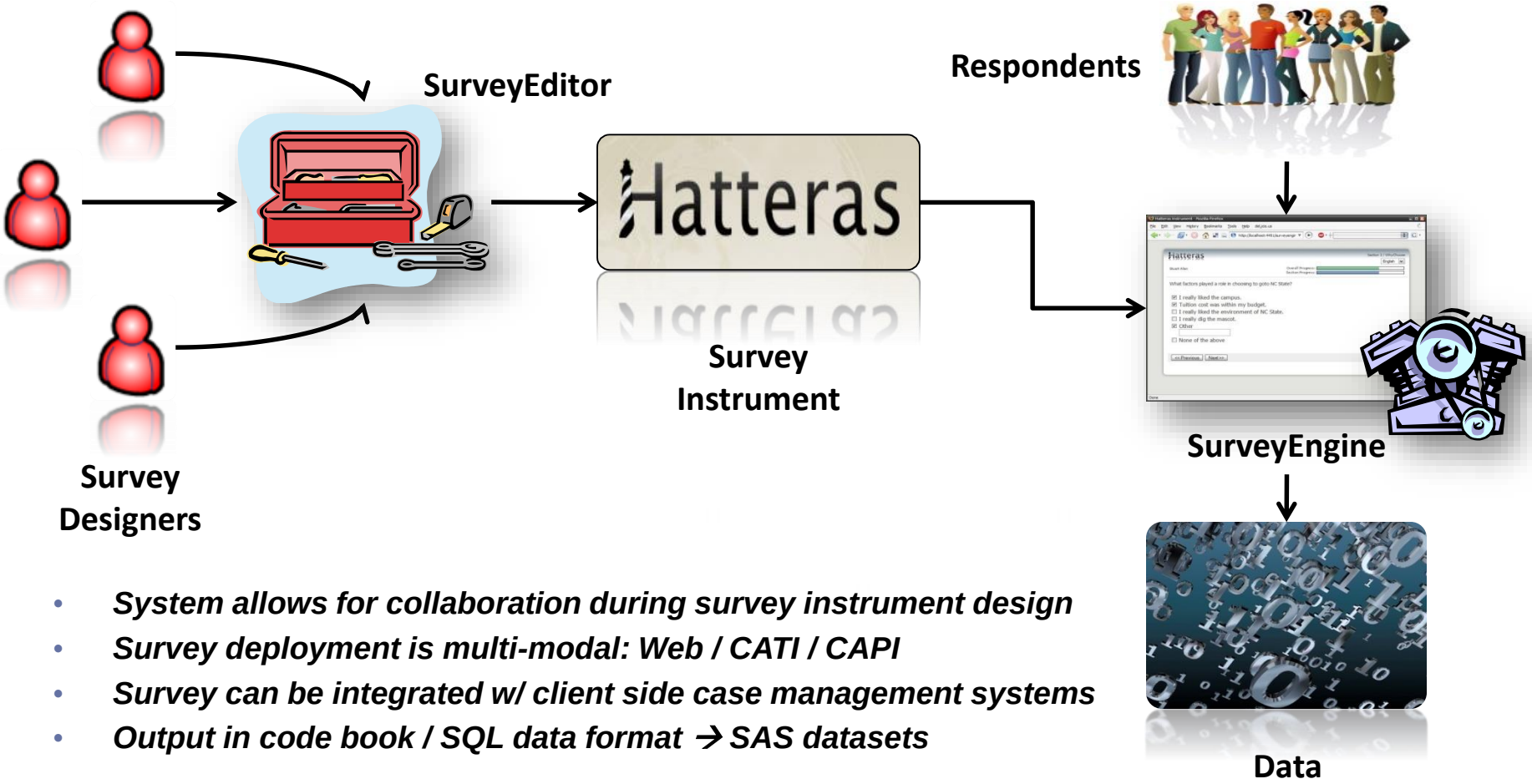
What is Hatteras?

- A standard development platform for surveys, initially developed for web-based surveys
- A platform shared by specification writers and programmers for instrument design and documentation
- A web based rendering engine based on common core libraries

Hatteras System Overview

- Hatteras = SurveyEditor + SurveyEngine + Utilities
- SurveyEditor
  - Edit instrument specs
  - Collaborate with team on specifications
  - Documentation utilities
- SurveyEngine
  - Website that runs the instrument
  - Based on Hatteras Core library
  - Testing utilities
- Utilities
  - Codebook generator
  - Blaise code generator

Hatteras System Overview



- *System allows for collaboration during survey instrument design*
- *Survey deployment is multi-modal: Web / CATI / CAPI*
- *Survey can be integrated w/ client side case management systems*
- *Output in code book / SQL data format → SAS datasets*

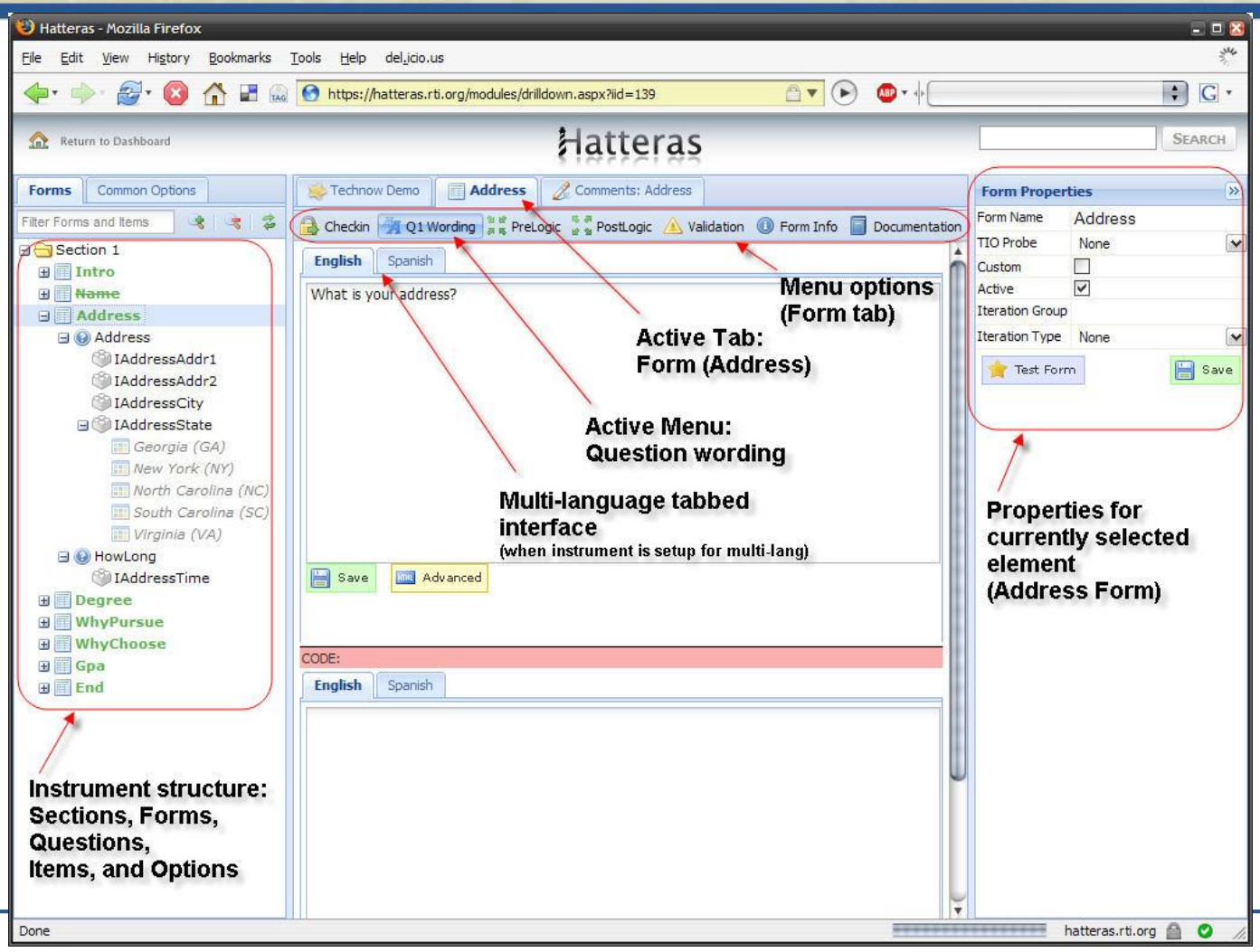
Challenge: Questionnaire Specification

Questionnaire development environment changes every time a new CAI software is introduced!

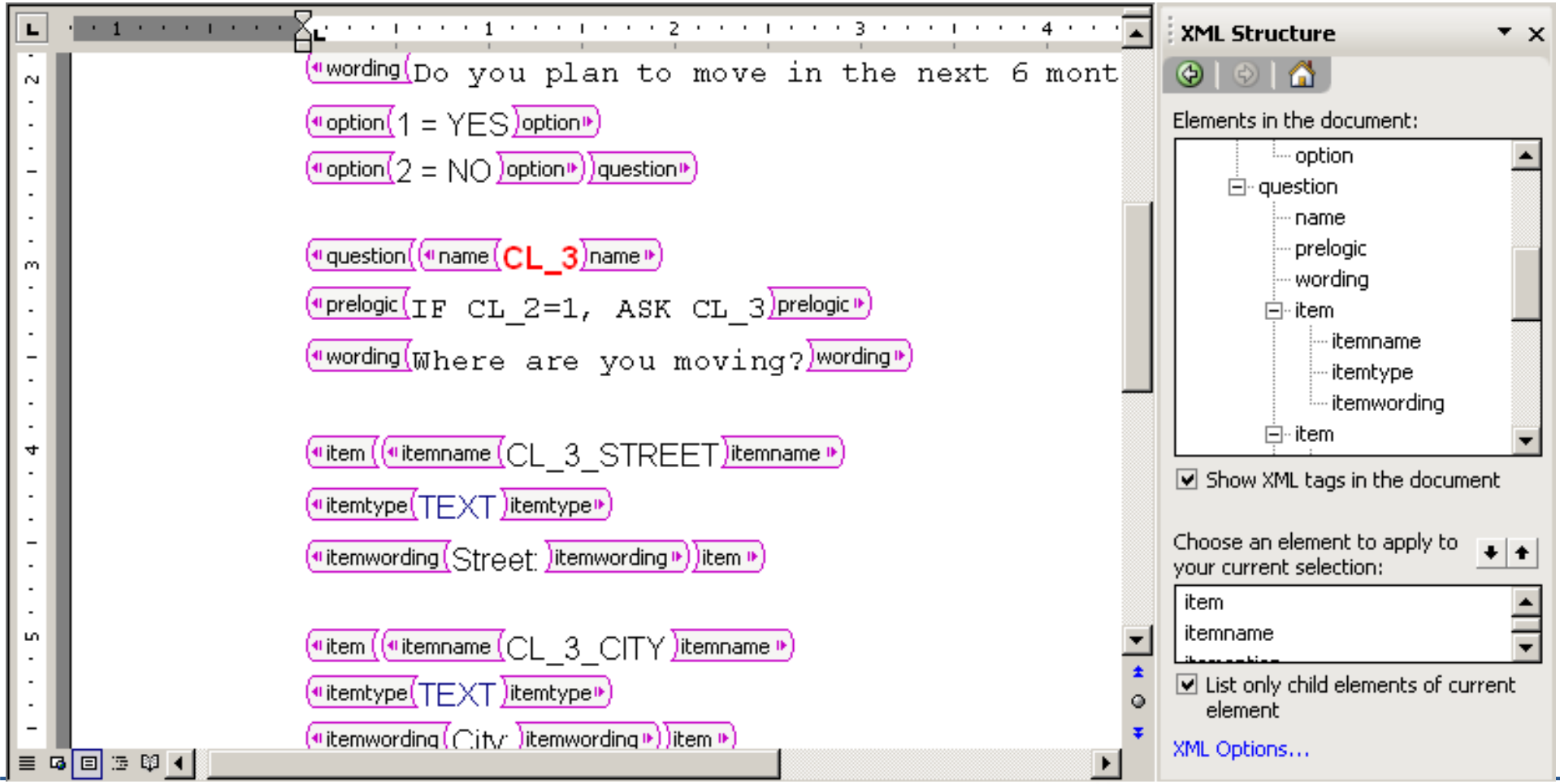
- Same editor (or IDE) is used by all users
  - Comments system facilitates collaborative development
- Same editor used for different modes and software (web, Blaise, data-entry)
- Specs can be provided in MS Word or Excel for block import into the system as a starting point
- Whole instrument can be copied over as a “template” especially for longitudinal studies using XML

Hatteras Dashboard

Hatteras Survey Editor



XML Tagging for Block Import



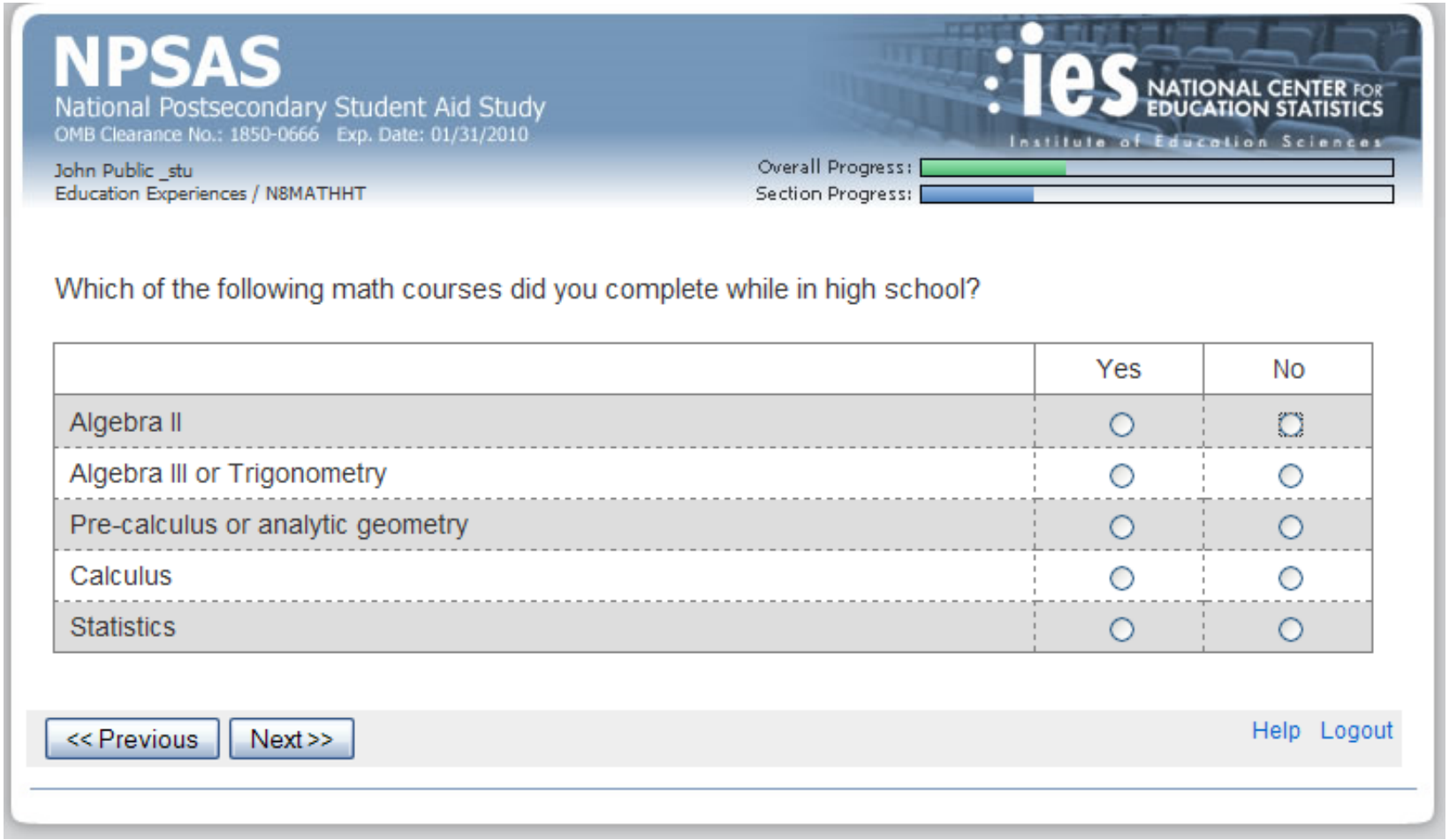
Challenge: Questionnaire Design

Web-based software tend not to support all question types and/or complex logic that are standard in client-based software

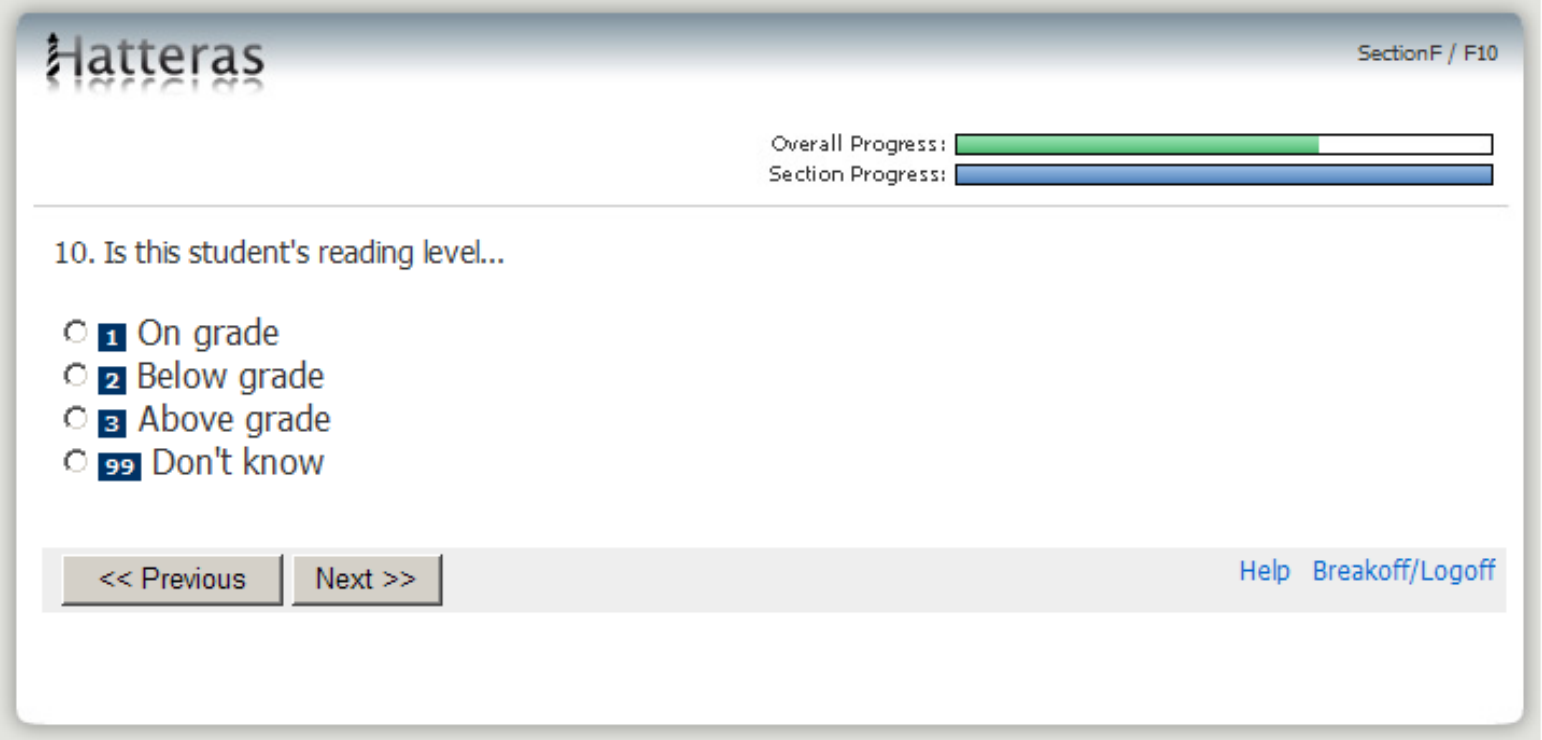
Features include:

- Standard question types as well as types such as conditional display for ‘specify other’
- Conditional display of question text
- C# language to specify skip logic and validation logic
- Looping through sets of questions
- Automatic flagging of backed-over items
- Look and feel for study specific “skin” can be customized easily
- Can add custom forms for unique scenarios
- Support for multiple languages

Hatteras Survey Engine



Hatteras Survey Engine With Keyboard Entry



Challenge: Mixed-Mode Studies

Spec it once, use it in different ways and oh yeah, make it compatible with other infrastructure systems

- Same Hatteras instrument can be set up for data collection in different modes: Self Interview, CATI, CAPI, or Data Entry
- Works seamlessly with RTI's CATI-CMS and IFMS for CATI and CAPI studies respectively
- Response options can be varied for different modes
- Keyboard entry including function keys (for DK/RF) makes it easier for interviewers
- Double-Key verification is built-in for data entry

Challenge: Non-linear Interviewing

Administering questions in sequence is so 20th century!

Need to be able to "jump-around" the instrument but still perform all of the validations

- Hatteras instrument can be administered in a non-linear manner (code named: Kangaroo engine) for data abstraction purposes
- Validations are deferred and implemented at section level
- Keyboard entry is still available
- Same IDE is used to specify the questionnaire

Challenge: Deployment

Can we host the survey in Timbuktu but still allow CATI interviewers to follow up?

- Develop the instruments in house but host them on the servers at client sites
- Synchronization tools facilitate updating of the instruments at client sites, FIPS-moderate environments and on CAPI laptops
- Version controlled common core library allows for new features to be added without affecting studies in production
- RTI's CATI-CMS can be used for follow up even if the study is hosted at client site or FIPS-moderate environments

Challenge: Performance

Everything is on the web fighting for bandwidth with Netflix but screen updates should match client-based software

- Use AJAX technology to refresh only portion of the web page that changes
- Common core library improves performance
- Application level caching: Single but a full copy of instrument loaded in memory serves all users and helps reduce server load
- Use Smartclient technology for off-line interviewing

Challenge: Data

Data has to be available almost immediately with codebooks that are self-explanatory

- Data can be extracted in unattended batch mode even from remote sites
- Built-in codebook generator includes question text, response options, and frequencies
- Paradata such as administration timing information are easily extractable
- Built-in modules facilitate generation of SAS datasets with all of the formats and labels

Future Challenges

In the future, Hatteras will be able to:

- Generate code for Handheld devices
- Exploit features of smart phones
- Provide a much more user-friendly and feature rich data dissemination system

That's all folks!

For additional information, please contact:

R. Suresh
Director, Center for Survey Technology
RTI International
919 541 6814
suresh@rti.org

Thank you!