

Blaise on Touch-Screen Tablets: the ELIPSS example

Alexandre Chevallier and Mathieu Olivier, CDSP – SciencesPo, France

1. Abstract

Starting in 2012, the ELIPSS project is a French probability-based web panel. All panel members are equipped with a touch-screen tablet to answer, each month, a self-administered survey programmed with Blaise 4, through a pre-install app.

This paper will describe, through some specific examples, how we had to overcome different challenges impacted by the specific technologies used, interaction between servers, and data collection.

Most of submitted surveys did not require specific developments, but some have instead required a lot of investment to adapt our solutions to Blaise.

We will present the solutions we found to address these problem, as for example for the development of drag & drop, clickable map, or the need to take pictures.

2. Introduction to the ELIPSS Panel

ELIPSS (Étude Longitudinale par Internet Pour les Sciences Sociales) is a probability based Internet panel, dedicated to Social Sciences, in similar way than the LISS Panel.

The pilot study is running since 2012 and consists in 1,039 panel members.

The target population is individuals living in private households in metropolitan France, aged from 18 to 75 at the entry in the panel, having sufficient command of the French language to answer self-administered questionnaires.

The surveys are conducted each month and it takes 30 minutes maximum to answer it. It could be on any topics. These surveys are designed by researchers and selected by the Scientific and Technical Board.

The covered topics are the ones of the projects selected by the Scientific and Technical Board. No topic is *a priori* excluded. In order to be eligible, surveys must have an exclusively scientific purpose. In other words, the purpose cannot serve the particular interests of public or private institutions.

One distinctive feature of ELIPSS is to equip all panel members with touch-screen tablets, in order to answer the self-administered surveys programmed with Blaise 4.8, through a pre-install app.

3. Tablets

Contrary to the other probability based Internet panels, Internet access is offered to all panel members, not only to those who do not have Internet access at home. Thus, each panel member receives a touch-screen tablet with a 3G Internet connection.

We chose tablets instead of other devices (as smartphones or PCs) because their interface is intuitive, providing simplified Internet access to people unfamiliar with new technologies, with a good agreement of the screen size (7 inches).

Due to, the design of surveys is exactly the same for all panel members equipped with the same device.

Moreover, mobile Web access gives more flexibility in completing surveys: panel members can choose their time and location to answer to questionnaires.

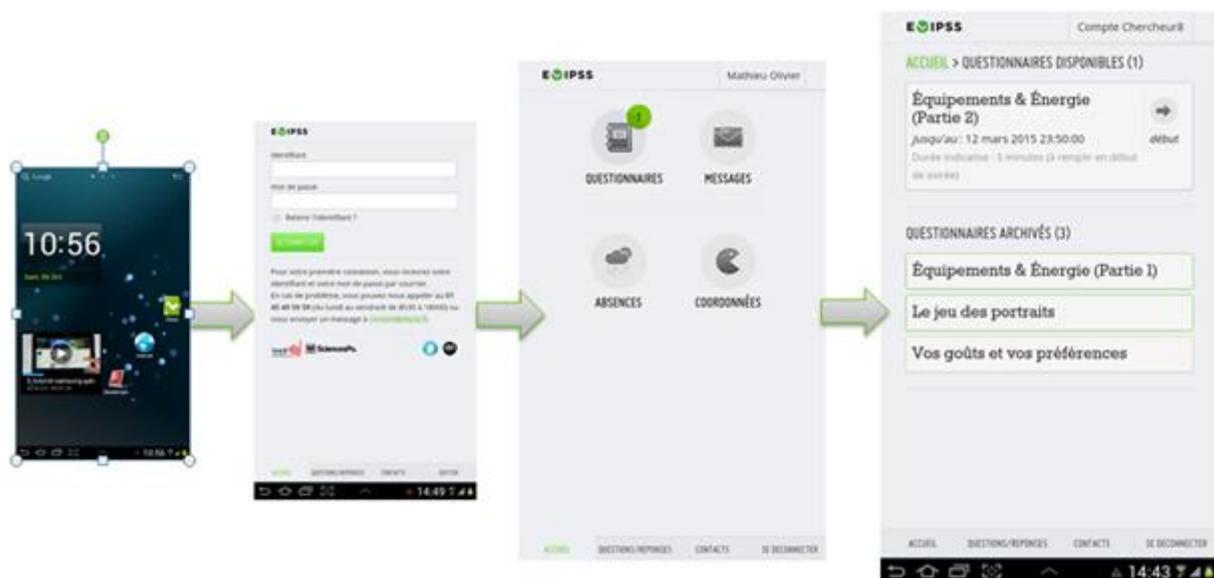
Besides the possibilities offered by the Internet (images, video...), the touch-screen tablet allows to administer innovating surveys, which is interesting for researchers and also motivating for panel members.

At last, in the Elipss project, there is no additional incentive each month. To equip all panel members with tablet which is considered as a material incentive. When we asked panel members to cite main reasons for which they agreed to participate in the Elipss panel, the touch-screen tablet was the primary motivation.

4. Blaise's integration

4.1 App

Figure 1. To the app to a questionnaire



The questionnaires are administered through a specific app which was internally developed and is designed for touch-screen device

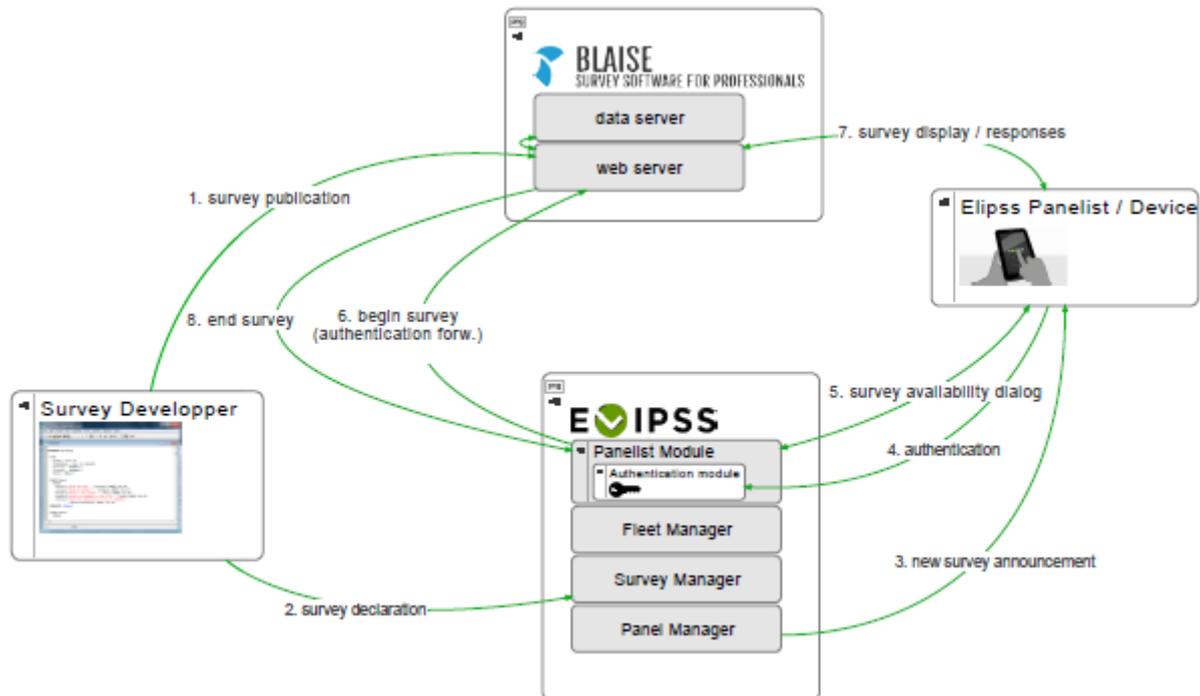
The screenshot displays the pre-installed (on each tablet) app on the home page of the tablet. The panelists have to log in to access the home page of the Elipss app (panelist module). The interface is simple. There are 4 icons : one to complete the questionnaires, another to send app messages to the panel managers, one to keep panel managers informed when panel members cannot answer the survey and the last one is to be used for any change in the panelist contact information (phone number, email address, etc.). FAQ, contacts and logout are at the bottom of the page.

When the tablet was delivered to the panel member's home, panelists were offered training by phone mainly on handling the tablet and exploring the Elipss applet. Two thirds of the panel members were trained, most of the others did not need to be trained. Trainers reported that 20% of panel members were not at ease with the tablet.

The first survey began with a tutorial part presenting different types of questions and the design of Elipss surveys.

4.2 Tools used

Figure 2. Technical Workflow



Through this diagram, we can distinguish different steps, from the survey's conception to retrieve data using Blaise software (version 4.8).

After programming, the survey is published on the Blaise server (1) and declared on the Elipss server (2). From Elipss server, survey is published on all our active panelists and they receive a notification on the Elipss application installed on the tablet (3). A message with a copy mail plus a SMS are sent to our panelists to inform them.

After the panelist's authentication (4) and the selection of a survey (5), Blaise server take over the communication (6). So panelists are able to answer the survey and the answers are saved in the database (workbdb) on the Blaise server after each page of the survey. Once the survey is finished, data are archived (so no more accessible) on the Elipss apps.

One of the difficulty was to create a new interface adapted with constraints of the tactile tablets. A stylesheet (.xsl) special for Elipss (first simplified version was kindly provided by LISS panel, thanks so much to them) is used in contrary of the default Blaise stylesheet. This allowed us to have a specific Elipss design (layout) for "classical" questions and questions adapted to the tablet (touch-screen keyboard, popup help, picker, etc.), but also some news types of question.

5. Adapting surveys with Blaise

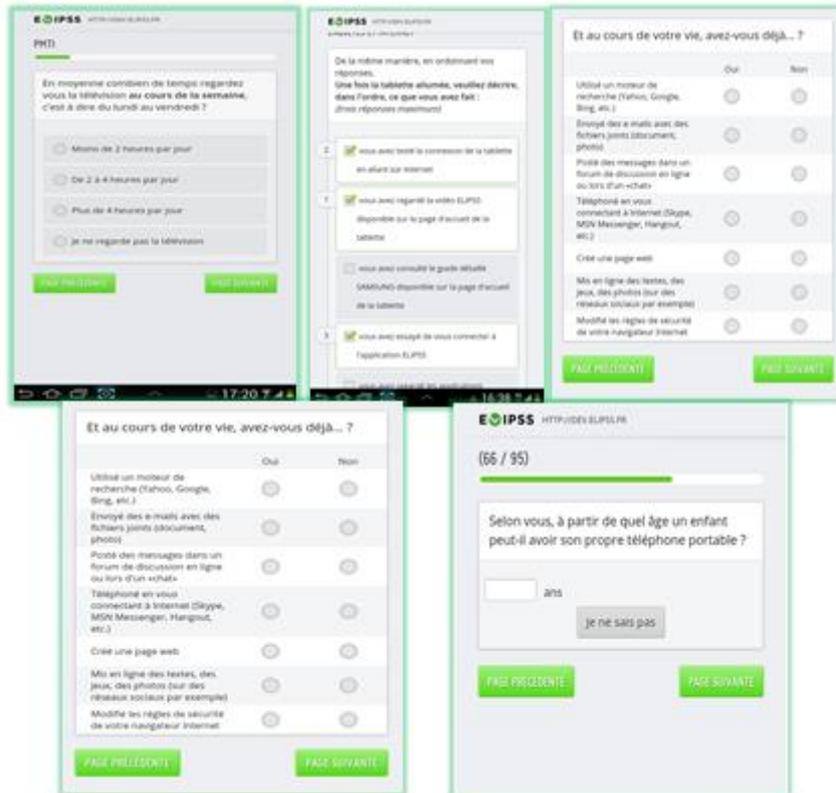
Contrary to Web surveys using different devices and different browsers, all the Elipss panel members have the same device and the same browser to answer questionnaires. This key design choice enables control over how questionnaires are displayed and aims for maximum measurement equivalence.

But we know that tablets also have limitations linked to connectivity problems and screen size.

The Liss panel team provided us their stylesheet to serve as a reference. So we were able to develop our own survey design, and to adapt field types as radio-button for enumeration type, checkbox for set

of type, sliders, group table, matrix, etc. We also designed field attributes like the non-response options refusal and dontknow as buttons when the option is allowed in the first time or after a reminder.

Figure 3. example of “classical” questions

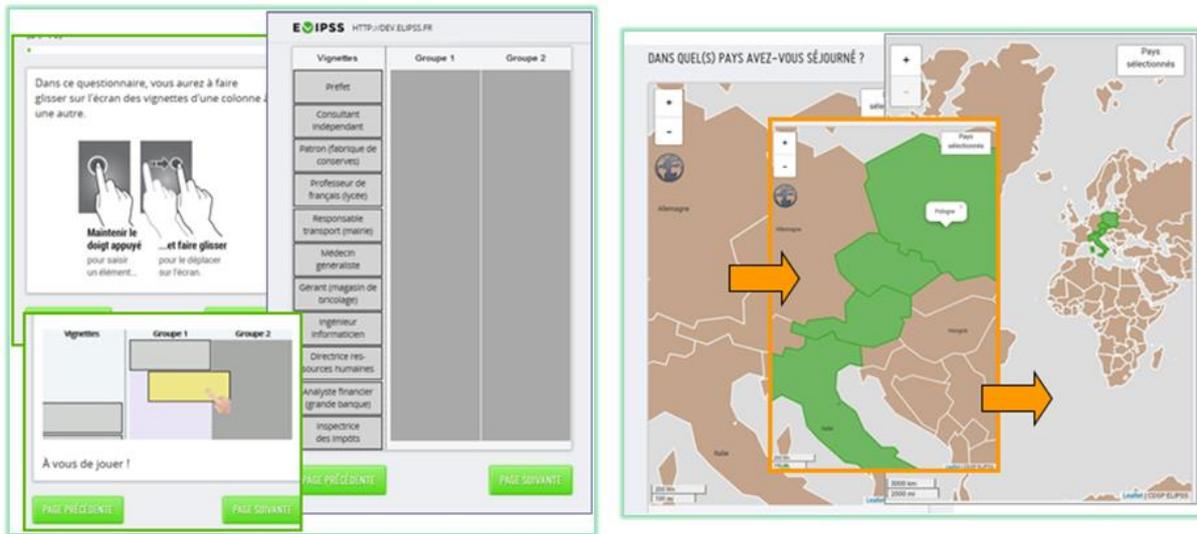


Another difficulty is to adapt “unclassical” questions.

Some surveys that requires more interactivity have provided the opportunity to develop specific features. This was only made possible mainly because all the panel members have the same equipment.

Then we used previous surveys functionalities like auto-recording with the microphone, drag and drop for an adaptation of an experimental survey on social stratification originally played with cards, interactive map to define residential strategies, etc.

Figure 4. example of “unclassical” questions: drag&drop and interactive map



We will see more on details how we developed these features through the photo’s example.

6. Specific feature: Photo example

To use all the functionalities of the tablet we added a Javascript layer composed with Vanilla, JQuery and Cordova. Cordova is a framework which allows to use the functionalities like camera, microphone or GPS on the tablet.

First of all, we call the library (Jquery and Cordova) inside the XSL file. Secondly, in the Blaise code survey manager, we add a special XML tag (<photo></photo>) in the specific question who will be parsed by our javascript.

Figure 5. xml tag

```
d7 (invisible) "<photo><folder>SHAMA_201607</folder><variable>d</variable><nb>5</nb><nb_max>6</nb_max><username>^uid</username></photo>
@BVeillez prendre en photo le ou les principaux espaces disponibles que vous utilisez pour cuisiner ou préparer des repas
@/@IMerci de ne prendre en compte que les surfaces qui sont dégagées de tout élément tel que plaques de cuisson, four, é
"Espaces préparation repas" : STRING, EMPTY
```

Indeed, our javascript layer parse the result from blaise and detect all the xml tag to initiate the different functionalities.

Then, the specific code linked with this tag will run and activate all the interface and code.

Figure 6. xml parser

```
ds.b.xml_question(
    "photo",
    [
        { 'name': "variable", 'expression': "photo > variable" },
        { 'name': "folder", 'expression': "photo > folder" },
        { 'name': "username", 'expression': "photo > username" },
        { 'name': "limit", 'expression': "photo > limit" },
        { 'name': "nb", 'expression': "photo > nb" },
        { 'name': "nb_max", 'expression': "photo > nb_max" },
    ],
    ds.b.utils.takephoto
);
```

Cordova camera function have two callback (success and fail), success callback give your image encoded in base64 in argument and allows you to manage easily the picture with vanilla javascript.

Figure 7. Cordova call

```
navigator.camera.getPicture(onPhotoDataSuccess, onFail, { quality: 50,  
  destinationType: Camera.DestinationType.DATA_URL });
```

Figure 8. JavaScript after Cordova

```
function onPhotoDataSuccess(imageData) {  
  // Uncomment to view the base64-encoded image data  
  
  // Get image handle  
  //  
  var photo = document.getElementById(photoFileName+'0');  
  var i=0;  
  
  while(photo != undefined){  
    i += 1;  
    photo = document.getElementById(photoFileName+i);  
  }  
  
  if(typeof(Storage) !== "undefined"){  
    var answer = document.getElementsByClassName('answer')[0];  
    localStorage.setItem(photoFileName+i, imageData);  
    var smallImage = document.createElement("img");  
    smallImage.id = photoFileName+i;  
    smallImage.className = "stored-photo";  
    smallImage.style.width = "400px";  
    smallImage.style.height = "600px";  
    smallImage.style.display = 'block';  
    smallImage.src = "data:image/jpeg;base64," + localStorage.getItem(photoFileName+i);  
  }  
}
```

Lastly to save the images, the panelist valid the question to go to the next one and we simply contact our server through an AJAX method with list of images to send them to the django server. At the time, Blaise server receive an answer “true” if the image was saved otherwise, it’s “false”.

7. Conclusion

Challenges are coming!

Since the beginning of 2016, 2,500 new panel members have been recruited to integrate the Elipss panel. This panel members are equipped with a new touch-screen tablet, with 4G connection.

The heterogeneity of equipments require a lot of work, in order to have surveys adapted to both devices and browsers.

Though “old” panel members recruited in 2012 will gradually receive the new tablet, we have to manage surveys on the 2 platforms for a while, as surveys continue to be published each month, waiting for the development of new features for future projects.