Information and Communication Technologies

EPIWORK

Developing the Framework for an Epidemic Forecast Infrastructure

http://www.epiwork.eu

Project no. 231807

D 5.5 Tests run in 2010

Period covered: Months 20th - 32th	Date of preparation: 30/09/11
Start date of project: February 1st, 2009	Duration:
Due date of deliverable: Month 32	Actual submission date: 15/10/11
Distribution: public	Status:

Project Coordinator: Alessandro Vespignani
Project Coordinator Organization Name: ISI Foundation
Lead contractor for this deliverable: AIBV

Work package participants

The following partners have taken active part in the work leading to the elaboration of this document, even if they might not have directly contributed writing parts of this document:

- AIBV
- ISI
- FGC-IGC
- CREATE-NET
- LSHTM
- KULeuven

Change log

Version	Date	Amended by	Changes
1	**30-09-10**		

Table of Contents

Test run in 2010

Introduction

The first objective of WP5 is to enhance the portability of the interactive Internet Monitoring Systems for diseases surveillance in Europe and to extend its implementation into five new countries: UK, Sweden, France, Spain and Germany.

At the beginning of the project, the system was active in four countries: The Netherlands, Belgium, Portugal and Italy.

The first year of WP5 was designed to be a pilot year including determination of a 'golden standard' set of ILI symptoms, database infrastructure preparation for, Internet database design, and translation of documents and testing. In that period, the surveillance activities of the already existing platforms have been intensified in the because of the 2009 H1N1pdm. All groups in the several countries where the IMS were not yet active, have accelerated the work and pushed forward the implementation of the surveillance system to start collecting data in real time during the newly announced pandemic.

At the beginning of the influenza season 2010-11, the surveillance system was active in The Netherlands, Belgium, Portugal, Italy and UK. Nevertheless, the existing platforms were still suffering of an excess of localization and there was the need of a more uniform and ready-to-deploy platform that would allow a more uniform data collection among the existing countries and a faster adoption by the new countries in the following years. During the spring and summer of 2010, the WP5 consortium worked at the development of a first prototype of the IMS that would have been deployed in the 'old' and 'new' countries during the influenza season of 2010-11. In particular, Sweden was ready to adopt the system, shortly followed by Germany, Austria and Switzerland.

IMS prototype and 2010 tests

At the end of the second year, the 'Gold Standard', developed in Task1 of WP5 was ready and the various teams collaborated to integrate the requirements posed by the Gold Standard into the IMS prototype in development. The general outline and implementation of *European IMS database infrastructure* and website (single centralized database, design of templates for the website in five new local versions) had proceed further during 2009 (see Task 2). The development of a common Gold Standard Questionnaire, the implementation of a unique database infrastructure where to collect all data coming from the different countries, and the availability of related documentation and information pages in different languages (obtained through Task 3), would allow researchers in WP5 to build a concretely portable system for surveillance that could be more easily exported and implemented in different countries.

In the design of the first prototype of the IMS that would unify the data collection across the several European countries, several aspects had to be taken into account. The implementation of the IMS in each country represented a separate scientific challenge and research problem. Each country has different population stratification and geographical distribution. The IMS implementation had to take into account these differences and design specific mapping/representation solutions as well as finding the optimal granularity for the data acquisition and statistical sampling.

The IMS first prototype was designed by the UvA subcontractor in the following way: the platform would consist of a Content Manager and a MySql Database for the local data collection. The IMS would have to fulfill a double necessity: it had to conjugate an easy management of content, news, info-graphics and it had to provide a user-friendly interface for the volunteers to access the surveys aimed at the collection of the epidemiological data. The data collection revolves around two main surveys that are proposed to the IMS users: a background or intake survey, proposed only once after the registration, and containing questions about the user's postal code, gender, birthdate, household, level of education, employment, chronic pathologies, smoking habits, etc. and a weekly survey with a list of symptoms from which the user would have to pick his/her symptoms corresponding to his/her health status.

For the content, news, info-graphics part, the Content Manager, based on Django CMS (https://www.django-cms.org/) was designed in a hierarchical way with the following features:

1. Arranged in a hierarchical way
2. First page in the first level of hierarchy is the main home page
3. Slug in the URL path
4. Each page can have its own template
5. The number of block in the page is determined by the template
6. Each content block can be filled by one or more content plugin
7. Content plugin is type of the content: text, file, picture, Google maps, etc.

The Content Manager would also have a News Journal that can be used to create contents that have data/time property and want to be shown in date/time based archive pages. For example: news pages, blogs, story etc. Each journal entry can be assigned into one or more categories, for an easier access to the archives of the news collected during each influenza season. The teams in the different countries could easily customize the templates and style from the front end since the management of these features was embedded in the back end platform and easily accessible also by non-Web experts. The platform would also be equipped with a "Reminder", i.e. a weekly email sent to users to remind them to fill the survey about the symptoms.

On the second aspect, i.e. the data collection, the complexity and richness of the 'Gold Standard' for the background and symptoms surveys required an ad hoc solution for the design of the surveys engine. The solution proposed in this first prototype was to develop a survey specification language, implemented as a Python script. In this Python script, each survey can be created as a Python class and each question is implemented as a python object that can be of several types (single choice, multiple choice, multiple choice, date input, text input).

To define a survey, a class that inherits d.Survey has to be created. More than one survey class can be defined but only the one with **Survey** name is used as the main survey class and therefore is used for the actual survey.

The survey class requires two fields:

- id is the survey id. This id is used in survey configuration and also in the data that will be sent to the central database.
- rules defines the question order/flow.

The question flow is defined in the rules field of the survey class. It is a Python set consisting of question classes. To create a conditional branch, a specially formatted Python dictionary instead of question class can be used. This Python dictionary contains **exactly** one element: a rule and a question set. The rule is written in a three elements Python set, each showing first operand, operator, and second operand, respectively.

Example:

```
{ (RegQ1, 'is', 1) : ( ... ) }
```

The above example means "If the value of RegQ1 **is 1**", then the following questions will be made visible.

Beside the actual value, there are three special operands for this branching definition:

- d.Empty is an empty value.
- d.Items(...) contains a set of values, e.g. d.Items(1, 2, 3)
- d.Profile(...) referes to the value of a question in the user profile pointed by the given question id, e.g. d.Profile('birthdate')

The available comparison operators are:

- is means both operands are the same.
- is-not means both operands are not the same.
- is-in means the first operand is one of the values given in the second operand. To use this operator, the second operand has to be a d.Items.

The question set defined in this branching definition is written in the same way as the main question set (rules) is written. Therfore, it may contain nested branches.

Example of a survey definition:

```
class Survey(d.Survey):
    id = 'dev-profile-0.0'
    rules = (
        RegQ1,
        RegQ2,
        RegQ3,
        RegQ4,
        RegQ5,
        RegQ6,
        { (RegQ1, 'is', 1) : (
            RegQ7,
        ) },
        RegQ8,
        RegQ9,
        RegQ10,
        RegQ11,
        RegQ12,
        RegQ13,
        RegQ14,
    )
class Survey(d.Survey):
    id = 'dev-survey-0.0'
    rules = (
        RepQ01,
{ (RepQ01, 'is-not', d.Empty) : (
   RepQ02,
   RepQ03,
   RepQ04,
    RepQ05,
   RepQ06,
   RepQ07,
   RepQ08,
   { (RepQ08, 'is-in', d.Items(1, 3)) : (
      RepQ09
   ) },
   RepQ10,
    { (d.Profile('RegQ5'), 'is-not', 1) : (
      RepQ11
   ) },
   { (d.Profile('RegQ6'), 'is-not', 1) : (
      RepQ12
   ) },
 ) }
```

This ad hoc survey specification language produces surveys that are then written as Python scripts. Each specification, i.e. each survey, has to be registered on the website and will then have its own identifier written inside the specification itself. To specify which configurations are then used for the surveys it is enough to modify the variables SURVEY_ID and SURVEY_PROFILE_ID in the platform settings.py file.

This approach, in principle very powerful and flexible, revealed to be not entirely user-friendly. In order to address the complexity of the ‘Gold Standard’ questionnaires, the python scripts underlying the surveys had to necessarily become quite complex and not very usable by non Python experts. Thus, at the beginning of the influenza season 2010-11 this issue still needed to be considered. The IMS platform needed to be tested to allow the assessment of software stability, database reliability and tune the advertisement campaign according to the various national habits and Internet penetration. Moreover, these requirements had to go along with plug-and-play features that would lessen the burden of deploying the platform in the new countries and would let the local teams concentrate on the Communication and Dissemination instead of having to deal with technical problems.

The various teams in the different countries in which the old platform was already present put a great effort in setting up the new one in time for the beginning of the influenza season, while in the new countries that were about to adopt the platform for the first time, like Sweden and Germany, the new system started to be deployed in October 2010 (see **Deliverable 6.2)**. The localization, translation, testing and fine tuning activities took place between October 2010 and February 2011.

In summary, the tests carried out during 2010 on the first prototype of the platform developed by the subcontractor UvA led the WP5 consortium to the conclusion that the approach of an ad hoc solution for the surveys coupled with a user-friendly Content Management system was a good solution even though the Survey Specification Language approach was quite difficult to maintain for the teams in the ‘old’ countries and not suitable for fast and ready deployments in ‘new’ countries willing to adopt the platform. Moreover, problems of data handling and visualization of results at different scale resolution still needed to be addressed and to conform to the local scenario: e.g. the choice between administrative regions and postal codes,

conformity of the choice made with the national privacy regulations, analysis of the hierarchical systems used by each country for administrative regions and/or postal codes, and others.
Along with the issues with the IMS prototype, the difficulties in managing the surveys software and the data collection software propagated to the centralized database that was still not able to collect the data from the different platforms in an unsupervised way. The infrastructure described in D5.1 for the EpiDB implemented by the UvA subcontractor was tested starting from October 2010 but the complexity and difficulties that the various team had in managing the Survey Specification Language approach made this solution not completely satisfying and not reliable for data collection from the various platforms in an unsupervised way.

Solutions to issues encountered during tests in 2010: IMS second prototype

To solve the issues encountered during the season 2010-11 with the first IMS prototype, the WP5 Consortium, in particular the ISI and IBV teams, has worked during the Spring and Summer of 2011 to develop a second prototype of the IMS that would maintain the combined approach of an ad hoc solution for the surveys and a user-friendly Content Manager based on Django CMS. In the second prototype, the software for the building, management, translation, export etc of the surveys (Surveys Editor) is completely integrated into Django CMS and can be administered from the Django back end in a much more user-friendly way that doesn't require expertise in Python programming. The new interface relies on an ad hoc software developed by the WP5 consortium to build surveys with a quite sophisticated logic flow without the need of programming skills, the interface is flexible and can be used to build new surveys and modify the already existing ones in a very rapid fashion, to suit the real time needs of this kind of platform. Moreover, the various problems of data handling and visualization of results at different scale resolution have been solved with a GUI (Graphical User Interface), integrated with the Surveys Editor, that lets the user generate charts and maps selecting SQL queries from the interface directly connected with the Database.

Edit workflow
----

1. The survey editor running on the client (Javascript and HTML/CSS) sends the questions, options and rules as XML to the server.

2. The server-side code extract relevant data from XML and fills Django models for questions, options and rules. All the interaction with the database is through Model instances.

3. When the survey is published the server-side code generates a dynamic Model based on survey structure and uses it to create the table where user responses will be collected.

Submission workflow
----

1. When the user accesses a survey the server-side code loads the questionnaire structure from the database using only Django Model instances and generates the corresponding HTML form.

2. When the user submits the data the server-side code loads the questionnaire structure from the

database, generates the same dynamic Model used in point (3) above and uses it to generate the Form instance that will be used to collect and validate POST data. If POST data passes validation the Form is used to fill the Model that in turn is used to store survey data to the database.

In general, all interaction with the database is done through the Django. ORM classes and no hand-written SQL is used in any case, letting to the ORM the task to bind databse backend drivers.

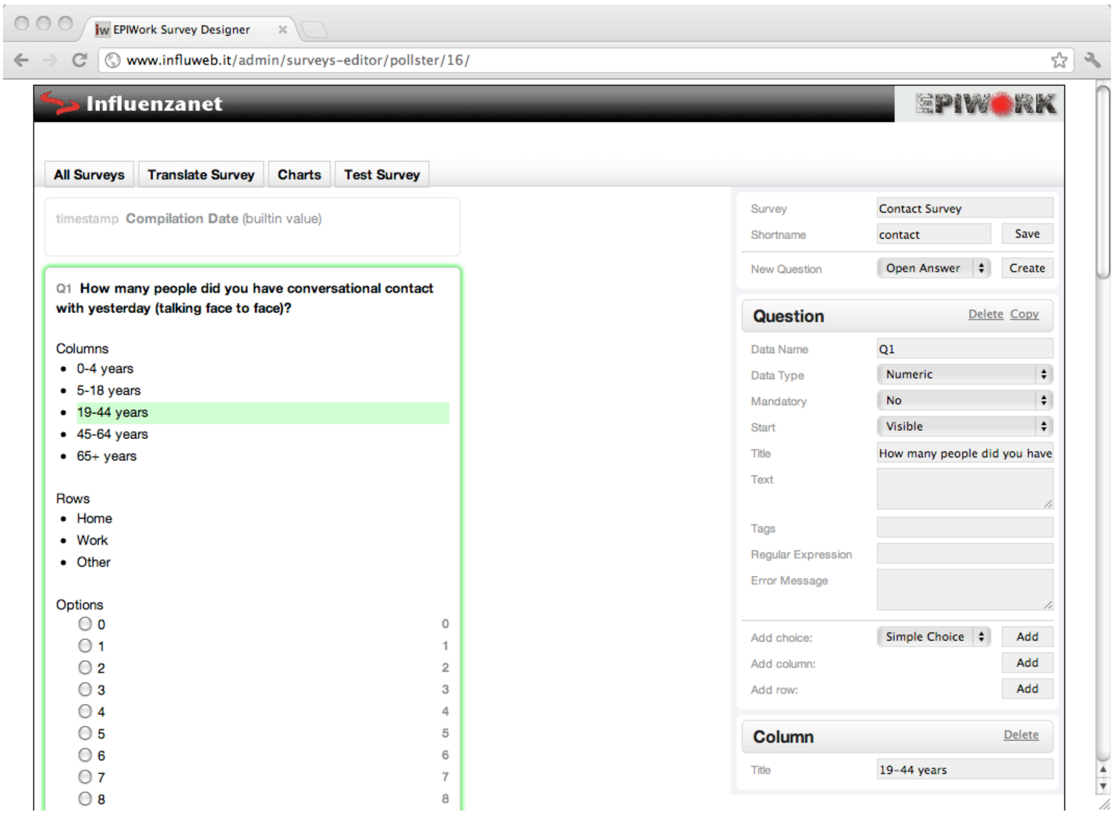
Charts

The administrator (and only the administrator) can create charts that execute arbitrary SQL statements (e.g., even INSERT, UPDATE and DELETE statements.)

The data generated by the query written by the administrator is replicated into a specific table (updated periodically) and used to generate the charts as seen by website users. Non-administrator users never access the original survey tables but only replicated data through a simple "SELECT * FROM <chart_table>" query.

The only external data passed to the chart query are the current user id and the current person gid: both are safetly passed using DBAPI bound variables.

This complexity is completely hidden for the administrator of the back end and quite complicated and sophisticated surveys, such as those described by the ‘Gold Standard’ paradigm, can be built by means of the GUI interface (see snapshot below)



timestamp Compilation Date (builtin value)

Q1 How many people did you have conversational contact with yesterday (talking face to face)?

Columns
- 0-4 years
- 5-18 years
- 19-44 years
- 45-64 years
- 65+ years

Rows
- Home
- Work
- Other

Options
○ 0 0
○ 1 1
○ 2 2
○ 3 3
○ 4 4
○ 5 5
○ 6 6
○ 7 7
○ 8 8

Survey: Contact Survey
Shortname: contact
New Question: Open Answer

Question
Data Name: Q1
Data Type: Numeric
Mandatory: No
Start: Visible
Title: How many people did you have
Text:
Tags:
Regular Expression:
Error Message:
Add choice: Simple Choice
Add column:
Add row:

Column
Title: 19-44 years

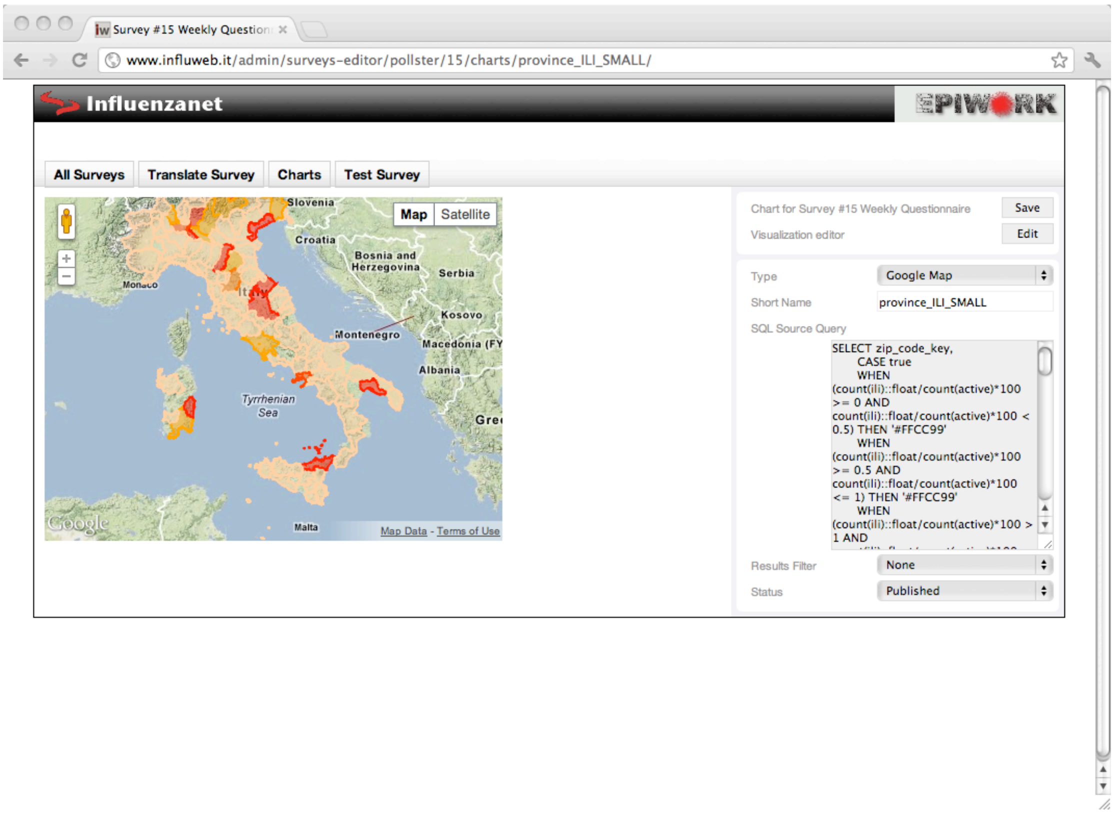


Chart for Survey #15 Weekly Questionnaire

Visualization editor

Type: Google Map

Short Name: province_ILI_SMALL

SQL Source Query

```
SELECT zip_code_key,
       CASE true
       WHEN
(count(ili)::float/count(active)*100
>= 0 AND
count(ili)::float/count(active)*100 <
0.5) THEN '#FFCC99'
       WHEN
(count(ili)::float/count(active)*100
>= 0.5 AND
count(ili)::float/count(active)*100
<= 1) THEN '#FFCC99'
       WHEN
(count(ili)::float/count(active)*100 >
1 AND
```

Results Filter: None

Status: Published

This approach conjugates the powerfulness and flexibility to plug-and-play features and has been successfully adopted by all the ‘old’ countries (The Netherlands, Belgium, Portugal, UK, Italy, Sweden) and has been deployed in France in December 2011 where, in less than two months, it has seen the enrollment of more than 4000 participants from all over the country. The system has been deployed also in Germany in February 2012 but the Communication campaign has not been equally successful.

For all the above-mentioned reasons, even the EpiDB has been re-implemented during 2011, as a collection of agreements/protocols and ready-made scripts, rather than as a full server stack based on REST technology. Specifically, agreements were made on a canonical form of the data to be exported from the various web platforms, the clients. Based on those agreements each client has its own script, which transforms the data. They may, due to local differences, be in a slightly different format to the canonical version. Key to this canonical data format is that it does not contain any information that can be directly used to identify users, i.e.: no email nor names. At the present moment, all individual Influenzanet partners have uploaded these canonized data to the central server (where each member country uses its own Unix-user) by using simple tools such as scp from shell. The data are loaded into a user-local database and aggregated and redistributed to the Influenzanet partners. We have run tests before and after downloading and uploading of data. The centrally collected data are displayed in the Influenzanet corporate website. Influenzanet.eu presents the project an its results in a reader-friendly way, in order to promote the concept of Internet-based Monitoring Systems in other countries, to expand scientific cooperation with colleagues all over the world and, last but not least, to show interested visitors what flu, vaccination and epidemiology is.

It contains an analysis of the collected Influenzanet data so far is presented, with as main and public elements:

- Graphs and data on flu and cold from all current (10) Influenzanet partners;
- the Netherlands, Belgium, Portugal, Italy, UK and France;
- Daily updates from all graphs and data;
- ILI curves, as compared to Google Flu, EISN, Temperature (also interactive);
- Curves of other syndromes and ILI curves within various subgroups;
- Participation data: participants, completed surveys, histograms;
- Week and incidence data, also as CSV files;
- All published articles on IMS and Influenzanet to date;
- A weekly update of modeling and flu surveillance news.

Conclusions

In conclusion, we can certainly affirm that all the issues encountered during the development, deployment and testing carried out during 2010-11 have been successfully overcome with the joint effort of the whole WP5 Consortium and that the second IMS prototype, extensively tested in 2011, is now the production platform adopted in all the participating countries.