MEDINFO 2019: Health and Wellbeing e-Networks for All
L. Ohno-Machado and B. Séroussi (Eds.)

doi:10.3233/SHTI190279

Model-Driven Architecture Based Software Development for Epidemiological Surveillance Systems

Azanzi Jiomekong[a,b], Gaoussou Camara[b,c]

[a] *University of Yaounde I, Cameroon*
[b] *Sorbonne Université, IRD, UMMISCO, F-93143, Bondy, France*
[c] *LIMA, University Alioune Diop de Bambey, Bambey, Sénégal*

Abstract

Epidemiological surveillance systems enable collection, analysis and dissemination of information on the monitored disease to different stakeholders. It may be done manually or using a software. Given the poor performances of manual systems, the software approach is generally adopted. Epidemiological surveillance systems are based on existing softwares, softwares developed from scratch given the specifications or softwares provided by a vendor. These solutions are not always suitable because epidemiological surveillance systems evolve quickly (new drugs, new treatment protocols, etc.), leading to software updates, which can take time (while waiting for a new version) and be expensive. In this article, we present the use of the Model-Driven Architecture (MDA) approach to model and generate epidemiological surveillance systems. The result is a complete MDA based methodology and tool to develop epidemiological surveillance systems. The tool was used to model and generate softwares that are now used for epidemiological surveillance of tuberculosis in Cameroon.

Keywords:

Epidemiological Monitoring, Software Design, Tuberculosis.

Introduction

Epidemiological surveillance systems enable collection, analysis, interpretation, and dissemination of health information to different stakeholders. These information are essential to the planning, implementation and evaluation of public health practices [1]. To strengthen epidemiological surveillance, additional activities are generally integrated. For example, in Cameroon, to efficiently fight against tuberculosis (TB), the Cameroonian National Tuberculosis Control Program (NTCP) in addition to epidemiological surveillance, manages anti-TB drugs, follow-up appointments of patients, sensitize patients, etc. However, these activities are done manually, causing problems like: low rates of promptitude and completeness, the production of basic statistics are generally late, insufficient sensitization of the population, difficulties in the geolocalization of hospitals so as to help patients get to the nearest hospital, and difficulties in managing lost patients (who did not come to take their treatment). The poor performances of manual systems have oriented the adoption of the software approach [2; 3]. For example, Blaya et al. [2] have proven that a web-based system to transmit laboratories reports decrease delivery time of laboratory results, reduce redundancy in resource utilization, and provide faster and more complete notification for public health purposes, decrease the number of reporting errors to Hcs, and improved monitoring of patients because clinicians have greater access to their history and laboratory data.

The rapid advancements and availability of health Information and Communication Technologies offer remarkable enhancement opportunities for epidemiological surveillance systems. The softwares used may be developed from scratch or existing options (e.g., District Health Information Software-DHIS [4], OpenMRS [5]) may be used. These solutions are not always suitable because epidemiological surveillance systems evolve faster (new drugs, new treatment protocols, etc.), leading to software updates, which can take time (while waiting for new version) and be expensive. On the other hand, depending on the data gathered on the field, the epidemiologist may need to collect a new parameter in order to explain a phenomenon (for example, the height of the patients in order to calculate their body mass index). This task may be done by using supplementary materials such as paper form or spreadsheet software (which can lead to a problem of data integration) or new requirements can be introduced and the software updated (which can lead to the problem of software regression). These problems can be mitigated if the system is designed using a well-defined framework or architecture permitting the rapid development and refinement of the surveillance software by non-informatics experts such as health workers. MDA has proven to be one of the best choices [6-10].

MDA provides methods and tools that can be used by domain experts (institutional operators with a deep knowledge in the domain) with limited high-level IT skills to build visual models (composed of graphical notations) and generate source code. Consequently, non-technical users can safely and effectively make changes to their software to reflect their changing needs and understanding of their business [8; 9; 11]. In the context of health informatics, there is a significant number of examples adopting MDA approach [6-13]: tracking patient information, data collection, mobile-health, Crisis and Emergency Management, etc.

In this article, we present the use of an MDA based approach for rapid development/update of epidemiological surveillance systems.

We applied the approach to build and epidemiological surveillance system within the EPICAM (Epidemiology in Cameroon) project. The project aims at improving epidemiological surveillance systems in Cameroon and is conducted by the National Tuberculosis Control Program (NTCP) in Cameroon in collaboration with the Unit for Mathematical and Computer Modeling of Complex Systems (UMMISCO), Centre Pasteur of Cameroon (CPC), and MEDES in France.

The rest of the paper is organized as follows: section 2 presents the MDA based approach; section 3 presents the results of the implementation of MDA for buiding the EPICAM platform; section 4 discusses the results; and section 5 concludes the paper.

Methods

Tuberculosis is a chronic infectious disease that kills almost two million people per year in the developing world [2]. To ensure high quality care is delivered in an efficient way, efficient information systems are essential. Then, the Cameroonian NTCP have planned a project called EPICAM with the goal to detect and treat TB patients and prevent disease from getting to people at risk (children who have been in contact with a patient or persons living with HIV). During this project, we have followed a set of principles, design activities and phases, based on agile software development and Model-Driven Architecture. In the next paragraphs, we will present the agile software development methodology, the MDA approach, and how we proceeded to develop epidemiological surveillance system in Cameroon.

Agile software development methodology

Agile software method relies on an iterative, incremental and adaptive development cycle which considers that the needs cannot be fixed and proposes to adapt the development to the changes. In a broader sense, agile method can be seen as a process consisting of an initialization step (the base version of the software is developed), an iterative step (new versions based on new specifications and users feedbacks are developed) and an incremental step (a set of new functionalities after an iteration is provided) [14; 15]. It's intended to support early and quick production of working code and allows fast deployment and adoption of health information [6].

MDA approach

MDA is an approach to software design, development and implementation which provides guidelines for structuring software specifications that are expressed as models. It focuses on forward engineering in which the executable source code is (semi)automatically generated from abstract, human-elaborated modeling diagrams such as a class diagram [6-8; 10; 12]. A class diagram describes the structure of a domain by identifying the domain classes (e.g., patient), their attributes (e.g., age, sex), their operations (e.g., calculate a body mass index) and the relationships amongst classes (e.g., the relation between a patient and his/her appointments at the hospital) [9; 13].

The process of building applications using the MDA approach can be summarized as follows: (1) the construction of a Computational Independent Model (CIM) which focuses on the environment in which the system will operate and its required features; (2) the construction of a Platform independent Model (PIM), which focuses on the aspects of the system features that are not likely to change from one platform to another; (3) the construction of a Platform Specific Model (PSM) which is obtained by integrating platform specific details to the PIM; (4) then, the PSM is converted into application code. It is the first and the second steps in the process that involves creativity and manual work; steps three and four are automated by the use of automated tools [10; 12]. The MDA approach is widely adopted to develop health information systems [6; 10].

MDA approach for epidemiological surveillance systems

Based on agile software development method and MDA, the methodology we propose for epidemiological systems development is composed of three main steps: pre-development, development and post-development.

Pre-development step: This step consists of the specification and the analysis of the system to be developed. During the pre-development step, the computer scientist works closely with the domain experts in order to make system specifications and analysis. The result is a first version of the software specification and analysis, containing sufficient information to develop the first version of the software.

Development step: This step is composed of two main phases:

- The first phase consists of the development of the first version of the software given the specifications and the analysis provided by the pre-development step. It proceeds as follows: **(1) Design:** The design consists to define the software architecture, choose the tools to be used to develop and deploy, and define the model (PIM) of the system in the form of a class diagram; **(2) Implementation:** The tools chosen in (1) will be used to transform the model into a platform specific model (PSM) which will be used to automatically generate the software.

  The result of this first phase is the baseline version of the software. The users will be trained on this basic version. Their use will generate feedback which will be very useful to continue the development.
- The second phase is an iterative and incremental phase in which each increment consists of the analysis of user feedback on the current version of software in order to add/remove functionalities. Each iteration proceeds as follows: **(1) Specifications and analysis:** the specifications and analysis are completed; **(2) Design:** the model (PIM) is updated according to new specifications; **(3) Implementation:** the model (PSM) is also updated and a new version of the software is generated. An increment is started after each iteration.

  The result of this second phase is a mature version of the software, tested and validated by the users.
- **Post-development step:** It consists of training the users on the use of the MDA tool to update the model and generate new versions of the epidemiological surveillance system. In fact, our goal is not necessarily to leave the software development to the domain expert, but to facilitate his/her task of the updating of the system to make it more close to his specifications without necessarily needing the help of an IT expert.

Development of the EPICAM platform

In this section, we present step by step how the MDA approach presented was used in order to develop the EPICAM platform for epidemiological surveillance of tuberculosis in Cameroon.

Specifications

To efficiently fight against tuberculosis, the Cameroonian National Tuberculosis Control Program has established a surveillance system through which it collects and shares data with the health professionals in health centers, health districts, health regions, the ministry of health, the general population and partners (Global fund, WHO, Centre Pasteur du Cameroun, etc.). However, this system is manually managed causing problems as previously introduced. Based on these problems, the specifications of an electronic system was developed.

To determine the specifications of the new system, we have collected information at each level (hospital, health district, health region and health ministry) of tuberculosis surveillance. These information were completed with the documents generally used for data collection and analysis. The desired system must permit users to: (1) collect, verify, synthesize data and make reports (weekly, monthly, quarterly) accessible at the district level, the regional level, and the central level; (2) follow patients and make SMS recall for those who did not come to an appointment; (3) manage anti-tuberculosis drugs so as to prevent stockouts; (4) locate the closest hospitals with respect to the location of patients; (5) sensitize the population by SMS; (6) permit the users to work offline and update their data when they connect to the network; (7) update the system each year by adding/removing some information on data collection supports and reports.

Analysis and design

The software analysis and design helps define an architecture, a tool based on MDA, and a data dictionary. **(1) The definition of the architecture:** The architecture presented by Figure 1 is the architecture of the system which suits an environment like Cameroon where access to Internet is not always available. This architecture is composed of a user component and a server component. The user component may be a web browser connected to the server via Internet. In this case, the client shares data with the server in a synchronized manner. In the case of the offline use of the software, the user component will be a desktop/mobile application with a local database. This application will be used by the users to share data with the server in an asynchronous manner. The synchronization module works as a mediator, which permits the server get new data from the desktop application local database and the desktop application get new update from the server; **(2) Choice of tools:** to develop the software, we have made a survey of MDA based generation tools used in the health domain. This survey helps identify tools such as Open Data Kit [16], Magpi [17] and Imogene [18]. ODK and Magpi are not suitable for us because the softwares generated run on mobile phones. Hence, we have chosen Imogene. Imogene is an open source platform developed by MEDES in France and used for the generation of data collection softwares. It provides graphical tools for creating models and generating tools that help generate applications for different platforms (Android, Linux, Windows, and MacOS). Taking advantage of the MDA approach, Imogene is used to update an already deployed application by updating the model, regenerating the new application and redeploying. The applications generated may be used in a synchronous/asynchronous manner. Imogene was used to generate data collection forms for the prevention and the follow-up of diabetes in France in 2009 and for data collection for tuberculosis surveillance in Georgia. By studying Imogene, we remarked that Imogene cannot completely solve the problem because it just generates data collection applications; **(3) definition of the data dictionary:** A data dictionary, containing the entities, their attributes, their relationships and a clear definition of each entity given by domain experts was established. The definition given by the experts will permit us to present these entities in the user interfaces. From this data dictionary, a class diagram (PIM) was built. This class digram was used to represent the previous entities, their properties and relationships in a graphical manner.

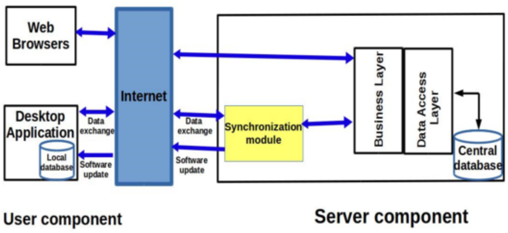

Figure 1– System architecture

Given the analysis and the *design* presented in the previous paragraph, we have developed EPICAM [19]. Based on Imogene, the EPICAM platform is an MDA based epidemiological surveillance system development. In addition to the data collection module provided by Imogene, EPICAM also integrates: **(1) reporting module:** for epidemiological report generation (in pdf format) using BIRT (Business Intelligence and Reporting Tool); **(2) geolocalization module** using OpenLayer, a library for creating interactive map on the Web; **(3) SMS module** for sensitization and patient recall; and **(4) managing drugs module** such as to prevent stockouts. EPICAM was developed to provide full software support for teams desiring to implement epidemiological surveillance system in their environment.

Implementation

Once the specifications, the analysis, and the design were completed, we have started the implementation. The EPICAM platform was therefore used to model (using visual notations) and generate a set of applications (presented in the results section). This first version was deployed in six hospitals in the two largest cities in Cameroon: Douala and Yaoundé. User feedback allowed us to complete the specifications (for example, integrate clinical radiology tests).

Iterations

After the development and the deployment of the software, the user feedback permitted us to complete the model and generate new versions of the applications. For users, everything is transparent, they simply discover new feature (for example, new field in a form).

After the application development, the NTCP selected twenty-five pilot hospitals for the deployment. These hospitals were selected given the quantity and the quality of the data they usually collect, and the ability of the health personnel to use computers. About fifty users were trained to use the softwares and three of them, from the central level were trained to use the EPICAM platform to update, generate and deploy a new version of the software.

Results

During the EPICAM project, we constructed a model, described in Figure 2, integrating the activities generally done by the NTCP. This model describes the entities of the domain, their attributes, relationships between entities and integrates the definition of each entities.

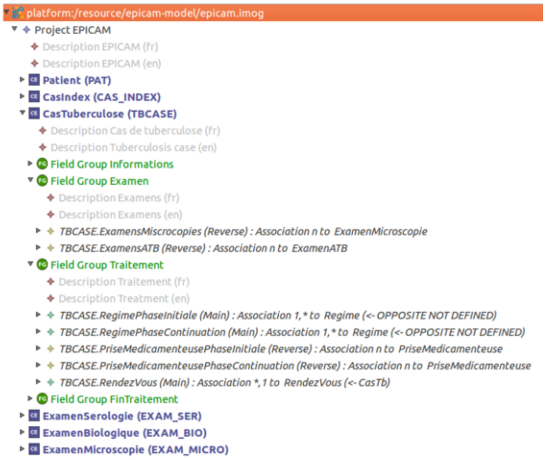

Figure 2– A model for tuberculosis surveillance software generation.

The model of Figure 2 permits us to generate a set of applications: **(1) Administration application:** used to manage health workers information, their roles, and their access rights on data; **(2) Web application:** used for epidemiological surveillance, manage anti-TB drugs, follow-up appointments of patients, sensitize patients, geolocalize the hospitals in order to orient patients. This application works in a synchronized manner. That is, when the health personnel using the application is connected to Internet; **(3) Desktop application** which has the same functionalities as the Web application, but is used in an asynchronous mode. That is, when Internet connection is not available, the system uses the local database as storage system and the local database is synchronized with the server when the Internet becomes availlable; and **(4) Synchronization application:** used to synchronize the client with the server (updating data or updating client applications).

Figure 3 presents an example of user interfaces completely generated using the model of Figure 2. This user interface is composed of the main user interface (entry point of the system) and a form used to register and follow patients (exams, drugs, appointments, etc.) during his/her treatment.

The system was deployed in twenty-five pilots hospitals, where the patients are generally treated for tuberculosis. We supervised the system in the course of the year 2015. Around 3900 (representing 15.6% of the annual number of TB cases in Cameroon) patients were registered and followed using the softwares. Given the success of this pilot phase, the NTCP have adopted the softwares generated as its electronic epidemiological surveillance softwares and extended it in ten regions, twenty new health centers in 2016 and 2017.

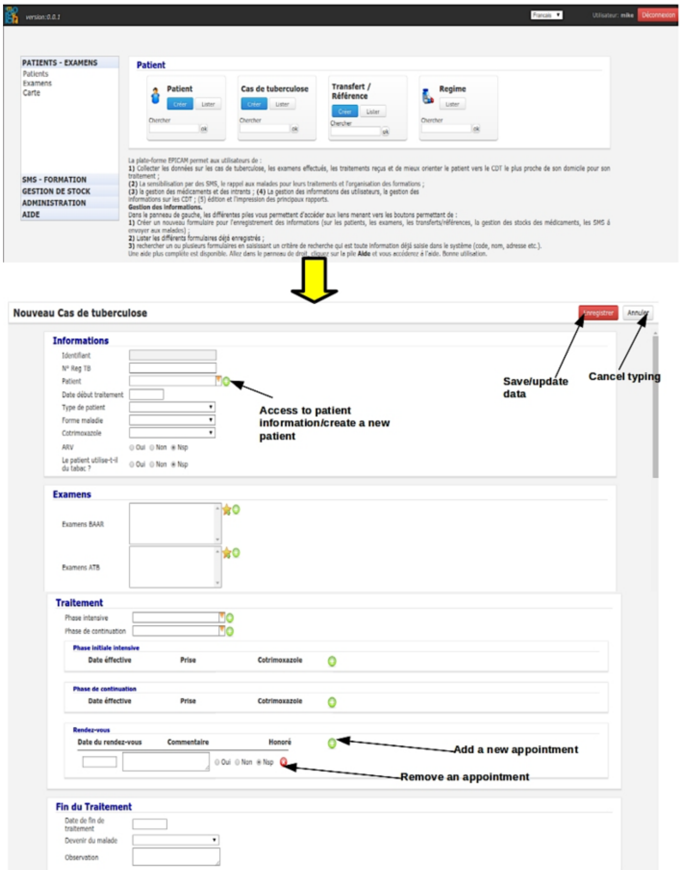

Figure 3– User interfaces completely generated using the model of Figure 2. It is composed of the software entry point and a form to register all the information during the patient treatment.

Discussion

During the EPICAM project, we produced a model for the generation of epidemiological surveillance system of tuberculosis. This model was used to generate a set of applications actually used for epidemiological surveillance of tuberculosis in Cameroon. In 2015, this platform proved its efficiency and has been adopted by the NTCP.

The use of agile software development methodologies to develop health information systems is not new. In fact, due to the complexity of the processes in health care, changes in the requirements may introduce a need for a correction of the implemented system, and can lead to regression [6]. This problem can be avoided by involving end users to the development process; and making an adapted methodology for iterative and gradual development. For example, Atanasovski et al. [6] have used an adapted agile methodology for the design and implementation of a health care information system used in Serbia and Macedonia. However, agile software development may have several drawbacks [15]. In the case of the EPICAM project, the experts were not available every time when needed. Then we were oriented to supplementary materials such as data collection forms, and statistics forms. These tools help us to advance in our job of modelling before user validation. During the validation, some elements not well modelled or defined were corrected by the users, which took more time and energy. Atanasovski et al. [6] show that agile methodology permitted fast deployment and adoption of health information but they need another approach (e.g., MDA) to assure its extensibility, soundness, interoperability and standardization.

To adopt the MDA approach, several conditions must be considered [9; 12]: (1) the project team must be experienced in modelling. If we take the example of the EPICAM project, during the training of end users in the use of the tool in order to update the model and generate a new version of the system, just one user from the ones trained to model and generate the software has accepted to use the tool; (2) have advanced tooling with appropriate modelling formalisms. The MDA doesn't work if each project is completely different. The time spent to get the model right and the transformation will be higher than working only with code. In the case of the EPICAM project, we have presented that before the adoption of the MDA approach, we have made a survey of tools and one tool was selected and adapted by adding new functionalities in order to fit to the software to be developed. This tool can be used to model and generate epidemiological surveillance systems of other diseases than tuberculosis. Then, this will lead to a positive return on investment.

Conclusions

In this article, we have presented a methodology based on MDA and using an agile methodology to develop epidemiological surveillance systems. This methodology was used to develop EPICAM platform now used for epidemiological surveillance of tuberculosis in Cameroon. During the pilot project in the year 2015, the system showed its efficiency and has been adopted by the NTCP in Cameroon. It should be noted that the methodology presented in this work can be used for the development of any health informatics.

In the future, we plan to integrate the SMS module [20] into the platform in order to permit villages in which Internet access is unavailable or damaged by a disaster to make their data available in time.

Acknowledgements

The EPICAM project was partially supported by grants from PRODESO (Programme Franco-Camerounais pour un Développement Solidaire).

We would like to acknowledge all the health workers for their feedback and constructive comments while using the EPICAM software. Our gratitude goes also to Dr. Nolna Désiré and Dr. Abena Jean Louis from the National Tuberculosis Control Program in Cameroon; Dr. Gaetan Texier from Centre Pasteur du Cameroon; Prof. Maurice Tchuente and Dr. Hippolyte Tapamo from UMMISCO in Cameroon; Mr. Laurent Braak and Mr. Julien Dupouy from MEDES in France and finally, all those who contributed to the EPICAM project.

References

[1] R.R. Frerichs, Epidemiologic surveillance in developing countries, *Annual review of public health* **12** (1991), 257-280.

[2] J.A. Blaya, S.S. Shin, M. Yagui, C. Contreras, P. Cegielski, G. Yale, C. Suarez, L. Asencios, J. Bayona, and J. Kim, Reducing communication delays and improving quality of care with a tuberculosis laboratory information system in resource poor environments: a cluster randomized controlled trial, *PloS one* **9** (2014), e90110.

[3] M.A. Mapatano and L. Piripiri, Quelques erreurs courantes d'analyse d'un système d'information sanitaire (RD Congo), *Santé publique* **17** (2005), 551-558.

[4] R. Dehnavieh, A. Haghdoost, A. Khosravi, F. Hoseinabadi, H. Rahimi, A. Poursheikhali, N. Khajehpour, Z. Khajeh, N. Mirshekari, and M. Hasani, The District Health Information System (DHIS2): A literature review and meta-synthesis of its strengths and operational challenges based on the experiences of 11 countries, *Health Information Management Journal* (2018), 1833358318777713.

[5] W.M. Tierney, M. Achieng, E. Baker, A. Bell, P.G. Biondich, P. Braitstein, D. Kayiwa, S.N. Kimaiyo, B.W. Mamlin, and B. McKown, Experience implementing electronic health records in three East African countries, in: *MedInfo*, 2010, pp. 371-375.

[6] B. Atanasovski, M. Bogdanović, G. Velinov, L. Stoimenov, D. Sahpaski, I. Skrceska, M. Kon-Popovska, D. Janković, and B. Jakimovski, Transforming an Enterprise E-Health System from Process Oriented to Model Driven Architecture, in: *7th International Conference on Information Society and Technology ICIST*, 2017.

[7] B. Atanasovski, M. Bogdanovic, G. Velinov, L. Stoimenov, A.S. Dimovski, B. Koteska, D. Jankovic, I. Skrceska, M. Kon-Popovska, and B. Jakimovski, On defining a model driven architecture for an enterprise e-health system, *Enterprise Information Systems* **12** (2018), 915-941.

[8] H. Schlieter, M. Burwitz, O. Schönherr, and M. Benedict, Towards Model Driven Architecture in Health Care Information System Development, in: *12th International Conference on Wirtschaftsinformatik*, 2015, pp. 497-511.

[9] J.M. Kanyaru, M. Coles, S. Jeary, and K. Phalp, Using visualisation to elicit domain information as part of the Model Driven Architecture Approach, in: *First International Workshop on Business Support for MDA*, 2008.

[10] W. Raghupathi and A. Umar, Exploring a model-driven architecture (MDA) approach to health care information systems development, *International journal of medical informatics* **77** (2008), 305-314.

[11] V. Curcin, T. Woodcock, A.J. Poots, A. Majeed, and D. Bell, Model-driven approach to data collection and reporting for quality improvement, *Journal of biomedical informatics* **52** (2014), 151-162.

[12] R. France and B. Rumpe, Model-driven development of complex software: A research roadmap, in: *Future of Software Engineering*, IEEE Computer Society, 2007, pp. 37-54.

[13] K.S. Rubin, T. Beale, and B. Blobel, Modeling for Health Care, in: *Person-Centered Health Records*, Springer, 2005, pp. 125-146.

[14] C. Larman and V.R. Basili, Iterative and incremental developments. a brief history, *Computer* **36** (2003), 47-56.

[15] D. Turk, R. France, and B. Rumpe, Limitations of agile software processes, in: *Third International Conference on Extreme Programming and Flexible Processes in Software Engineering*, 2002.

[16] W. Brunette, M. Sundt, N. Dell, R. Chaudhri, N. Breit, and G. Borriello, Open data kit 2.0: expanding and refining information services for developing regions, in: *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, ACM, 2013, p. 10.

[17] Magpi, Home, https://home.magpi.com/, 2019. Accessed April 1, 2019.

[18] MEDES-IMPS, Welcome to imogene, https://github.com/medes-imps/imogene/wiki, 2019. Accessed April 1, 2019.

[19] UMMISCO, EPICAM Platform, http://github.com/UMMISCO/EPICAM, 2019. Accessed April 1, 2019.

[20] A. Jiomekong, L. Broto, D. Hagimont, S. Temate, and M. Tchuente, Data collection in a degraded network: case of developing countries or countries in crisis, in: *Proceedings of the 6th International Conference on Theory and Practice of Electronic Governance*, ACM, 2012, pp. 406-409.