Integrating Sentinel Surveillance Systems and Electronic Health Records

The case of Médicos Sentinela and the National Health Service

Hugo Mendes Paixão

Thesis to obtain the Master of Science Degree in

Biomedical Engineering

Supervisors:
Professor Mário Jorge Costa Gaspar da Silva
Doctor Ana Paula Ambrósio Rodrigues

Examination Committee
Chairperson: Professor Luís Humberto Viseu Melo
Supervisor: Professor Mário Jorge Costa Gaspar da Silva
Member of the Committee: Doctor José Augusto Rodrigues Simões

June 2015
Time you enjoy wasting, was not wasted.

Marthe Troly-Curtin
I would like to express my most sincere thanks to everyone that helped and supported me during the elaboration of this work.

Firstly, I would like to thank my advisors professor Mário Silva and Ana Paula Rodrigues for the priceless guidance. Their knowledge and insight were crucial to the development of this thesis.

I express gratitude to the Department of Epidemiology at INSA for welcoming me in their “home” during the time of this project.

I also thank all the physicians in Médicos Sentinela network, for their valuable support and feedback regarding this project, but above all, for their invaluable effort to improve our community health.

I would like to acknowledge MedicineOne for their collaboration in this project, particularly to engineers João Miguel, João Sousa and Marco Tinoco, who took the time to guide me through MedicineOne system.

Additionally, I am grateful I had my dearest colleagues and friends David Machado, João Ramalhinho, Ricardo Filipe and Tiago Chaves by my side during my academic journey in IST, through the good and the bad moments.

Finally and above all, I am forever thankful to my family for their unconditional support and love, not only during my academic journey but during all my life.
Abstract

In Public Health, sentinel surveillance is an activity in which a selected number of health professionals provide regular reports on a set of health problems. Initial sentinel surveillance systems used paper forms as the reporting tool. More recently, web-based applications with manual data entry took over. However, the adoption of electronic health record (EHR) technology provides an emerging source of data to be tapped for surveillance reporting. This dissertation proposes an approach for the integration of a sentinel surveillance system and EHR systems, which was developed for the Portuguese sentinel network Médicos Sentinela (MS). It includes a tool for detecting notifiable cases in an EHR and extracting the relevant data to automatically fill MS notification forms. Physicians can then validate and complete the notification forms if necessary. An initial implementation of the tool was used to validate the proposed approach, showing that it could correctly fill automatically almost all of the notification forms. A preliminary assessment with a group of sentinel physicians has also shown that user satisfaction was high, with the tool greatly simplifying and reducing the time spent on the notification process.

Keywords

Sentinel Surveillance, Public Health and Clinical Systems Integration, Public Health Informatics, Médicos Sentinela Network
Resumo

Em Saúde Pública, a vigilância sentinela é uma actividade na qual um grupo restrito de profissionais de saúde submetem regularmente notificações relativas a um conjunto de indicadores de saúde. Originalmente preenchiam-se formulários em papel para fazer as notificações, mas recentemente a introdução directa dos dados em formulários Web passou a ser preferida. Contudo, os Registos Clínicos Electrónicos constituem hoje uma fonte emergente de dados, que podem ser utilizados para fazer notificações em sistemas de vigilância sentinela. Esta dissertação propõe uma abordagem para a integração de um sistema de vigilância sentinela e um sistema de Registos Clínicos Electrónicos, que está a ser desenvolvida para a Rede Médicos Sentinela (MS). Foi desenvolvida uma ferramenta para detectar casos passíveis de notificação num sistema clínico e extrair os dados necessários ao preenchimento de formulários de notificação da Rede MS. Os médicos sentinela validam e completam a informação extraída, se necessário. Uma implementação inicial desta ferramenta foi utilizada para validar a abordagem proposta, demonstrando a viabilidade de extrair grande parte dos dados necessários ao preenchimento das notificações da Rede MS. Uma avaliação preliminar com um grupo de médicos sentinela evidenciou ainda que a satisfação dos utilizadores era elevada, com a ferramenta a simplificar e reduzir o tempo necessário para o processo de notificação.

Palavras Chave

Vigilância Sentinela, Integração de Sistemas Clínicos e de Saúde Pública, Informática em Saúde Pública, Rede Médicos Sentinela
Contents

1 Introduction ................................................................................................. 1
  1.1 Motivation ............................................................................................. 2
  1.2 Goal ....................................................................................................... 2
  1.3 Results .................................................................................................. 3
  1.4 Methodology ......................................................................................... 4
  1.5 Dissertation Outline ............................................................................. 5

2 State of the Art ........................................................................................... 7
  2.1 Public Health ......................................................................................... 8
  2.2 Public Health Surveillance ..................................................................... 9
  2.3 Sentinel Surveillance ............................................................................ 9
    2.3.1 Definition and Historical Perspective .............................................. 9
    2.3.2 Importance on Influenza Surveillance ............................................ 10
      2.3.2.A European Influenza Surveillance Network (EISN) ..................... 11
      2.3.2.B U.S. Outpatient Influenza-like Illness Surveillance Network (ILINet) .... 12
    2.3.3 Sentinel Surveillance Information Systems .................................... 13
  2.4 Electronic Health Record ....................................................................... 14
    2.4.1 EHR usage in Europe ................................................................. 15
    2.4.2 EHR in the NHS ........................................................................... 16
      2.4.2.A SAM ......................................................................................... 16
      2.4.2.B MedicineOne ......................................................................... 16
    2.5 Summary .......................................................................................... 17

3 Médicos Sentinela ....................................................................................... 19
  3.1 Overview ............................................................................................ 20
  3.2 Notifiable Health Events ...................................................................... 22
  3.3 Notification Forms .............................................................................. 22
    3.3.1 General fields .............................................................................. 23
    3.3.2 Influenza-like Illness Specific Fields .............................................. 24
    3.3.3 Oral Anticoagulants Prescription Specific Fields ......................... 24
    3.3.4 Diabetes Specific Fields .............................................................. 25
    3.3.5 Hypertension Specific Fields ......................................................... 25
# List of Figures

1.1 Conceptualization of Sentinela+ system. .................................................. 4

2.1 Ten essential services of public health .......................................................... 8
2.2 Example of a plot using data from EISN ......................................................... 11
2.3 Examples of plots using data from ILINet ...................................................... 12

3.1 Geographical distribution of sentinel physicians in MS .................................. 20
3.2 Screenshot of MS-RIOS application web form interface, showing notification forms general fields ................................................................. 27
3.3 Screenshot of MS-RIOS application web form interface, showing notification forms hypertension fields ......................................................... 27
3.4 Screenshot of MS-RIOS application, showing the interface that allows public health staff export data from the notification forms ......................................................... 28
3.5 Notification process in MS network ................................................................. 29

4.1 Comparison between the current system and the proposed system .................. 32
4.2 Conceptualization of Sentinela+ system ......................................................... 33
4.3 Main window in Sentinela+ client application ................................................ 34
4.4 Basic structure of a notification form in Sentinela+ .......................................... 35
4.5 Interface of the Connector module, used to request data extraction and handle the response ................................................................. 39
4.6 Interface to manage cases and interface of an automatically filled form .............. 39
4.7 Interface to request data sending to Sentinela+ server .................................. 40
4.8 Semi-automated reporting process in Sentinela+ client application .................. 40
4.9 Manual reporting process in Sentinela+ client application ....................... 41
4.10 Interface to notify a period of time without notifications ............................. 42
4.11 Warning messages in Sentinela+ ................................................................. 43
4.12 Error messages in Sentinela+ ................................................................. 44
4.13 Client-server communication, between Sentinela+ client and Sentinela+ server 45
4.14 Screen shot of the application developed for data extraction ..................... 46
4.15 Conceptualization of Sentinela+ system, with a connection to MedicineOne 47
4.16 Tables and fields used to extract data and the relationships between the tables 49
List of Tables

3.1 General fields in notification forms .................................................. 23
3.2 Influenza-like illness fields in notification forms ............................... 24
3.3 Oral anticoagulants fields in notification forms ................................. 24
3.4 Diabetes fields in notification forms .................................................... 25
3.5 Hypertension fields in notification forms ........................................... 25
3.6 No cases and absence fields in notification forms ............................... 26

4.1 NOTIFICATIONS table fields, data input types and description of each field .............................................................. 37
4.2 ICPC2 codes for the MS-notifiable health events ............................... 49
4.3 ATC codes for the MS-notifiable anticoagulants ................................ 49
4.4 Mapping between data elements in Sentinela+ NOTIFICATIONS table and data elements in the MedicineOne EHR. This mapping allows to determine which fields in MedicineOne can be used to fill in Sentinela+ notification forms .............................................................. 50
4.5 Mapping between MedicineOne database fields and XML response markups for ExtractDisease web method ........................................... 53
4.6 Mapping between MedicineOne database fields and XML response markups for ExtractAnticoagulant web method ................................ 53

5.1 Evaluated factors in each interface element ...................................... 59
Abbreviations

ACSS  Associação Central de Sistemas de Saúde
ARI  Acute Respiratory Infection
CDC  Centers for Disease Control and Prevention
CSV  Comma Separated Values
DBMS  Database Management System
EHR  Electronic Health Record
EISN  European Influenza Surveillance Network
EU  European Union
GP  General Practitioner
GUI  Graphical User Interface
ILI  Influenza-like Illness
ILInet  U.S. Outpatient Influenza-like Illness Surveillance Network
INSA  Instituto Nacional de Saúde Dr. Ricardo Jorge
MS  Médicos Sentinela
NHS  National Health Service
QUIS  Questionnaire for User Interaction Satisfaction
SAM  Sistema de Apoio ao Médico
XML  Extensive Markup Language
1 Introduction

Contents

1.1 Motivation .................................................. 2
1.2 Goal ......................................................... 2
1.3 Results ....................................................... 3
1.4 Methodology ............................................... 4
1.5 Dissertation Outline ...................................... 5
1.1 Motivation

In public health, sentinel surveillance is a type of surveillance in which a limited network of health care professionals voluntarily reports data on a set of health problems. Sentinel surveillance can supply a sample of information to public health decision makers when data on the entire population are not available. Additionally, it allows collecting high-quality, detailed data about particular health conditions, which is not possible through passive surveillance systems, such as reports on notifiable diseases required by law.

Sentinel systems are usually based on data collection by a selected group of General Practitioners (GPs), although they sometimes also include physicians from other specialities. Data are collected through structured notification forms. In state-of-the-art sentinel networks, data are manually entered in web forms and transmitted to central databases through secure internet connections. Nonetheless, many networks are still paper-based or use both paper and the Web.

In their activity, physicians already dedicate a substantial amount of work time to data entry. This has posed a problem to physicians since the introduction of computers in health care. Notification forms in sentinel networks are manually filled, requiring an additional data entry process and consuming even more time. Besides, most of the data entered in notification norms is duplicated, which makes this process a repetitive task.

However, advances in information technology, along with the increasing adoption of the Electronic Health Record (EHR) as a standard in health care, provide an opportunity to facilitate the exchange of information between public health and clinical care. Since EHR systems capture the whole spectrum of medical data, it seems advantageous to use these data for automated filling of sentinel surveillance notification forms.

The opportunity to conduct a study regarding the use of EHR data for automated reporting in a sentinel network motivated this project, developed in the context of the Portuguese sentinel network, Médicos Sentinela (MS), in collaboration with the Department of Epidemiology of the Instituto Nacional de Saúde Dr. Ricardo Jorge (INSA), which coordinates the network. Created in 1989, MS allowed the surveillance of 50 health conditions ever since, including systematic continuous surveillance of diseases such as Influenza-like Illness (ILI), hypertension and diabetes, and contributing in an unique way to the estimation of incidence rates of many health problems with a strong impact on the Portuguese population.

1.2 Goal

This dissertation addresses the development of an architecture for a physician-supervised process for automatic generation and submission of notification forms to a sentinel network and implementation of the software to support it. The two main goals were:

1. To research the viability of using the data in EHRs to avoid duplicate data entry processes when providing data to a sentinel network.
2. Develop and evaluate a system integrating MS network and EHR systems, using one of the EHR systems of the National Health Service (NHS).

In MS, similarly to most sentinel networks, data collection is performed through web and paper forms. The core idea of this project was to automate the filling of MS notification forms by detecting cases of notifiable conditions from the EHR, and extracting all the necessary data regarding these cases to fill notification forms. Furthermore, EHR data must be de-identified before they are sent to MS server, where they can be accessed by public health professionals.

The approach used in this project to integrate MS with an EHR system used in the NHS should provide useful insights regarding how can other sentinel surveillance and EHR systems be integrated.

1.3 Results

I implemented the proposed system, called Sentinela+, together with an interface to the MedicineOne EHR system, to extract data for cases of MS-notifiable health conditions.

Sentinela+ includes a desktop client application, which communicates with MedicineOne server over the network, through HTTP protocols. To bypass security firewalls, I also developed a Web service to be hosted in MedicineOne server. This service is able to query the MedicineOne database about notifiable events and return clinical data back to clients. The Sentinela+ client application can automatically fill most of MS notification forms using these data and send them to a server that I have set up for further processing, where they can be accessed by the MS system administrators. The conceptualization of the whole system is illustrated in Figure 1.1.

A sample of MS sentinel physicians tested the system and evaluated it through a user satisfaction questionnaire. Feedback received was positive. Users rated how the system impacted the notification process in two aspects: facilitation of the notification process (average score: 8,0 out of 9) and reduction of the time necessary to fill notification forms (average score: 7,3 out of 9). Additionally, users also rated four user interface aspects: screen (average score: 7,3 out of 9), terminology and system feedback (average score: 7,9 out of 9), learning (average score: 8,1 out of 9) and system capabilities (average score: 8,5 out of 9). The system was also subjected to performance tests, which demonstrated that response times were acceptable: in the order of 50ms with 1 user request for 10 cases data extraction and in the order of 400ms with 100 sentinel users requesting a data sending process.

The Sentinela+ prototype is fully functional and ready for deployment in primary care centres that use MedicineOne EHR system, to extract real patient data. Moreover, Sentinela+ stand-alone desktop application provides an alternative reporting method to existing web based interfaces and paper reporting ready to be used.

Given the successful integration of MS sentinel surveillance system and MedicineOne EHR system, this development could serve as a starting point for integrating other sentinel networks with EHRs.
1.4 **Methodology**

The development of this dissertation followed this methodology:

1. Literature study on relevant concepts, such as public health, public health surveillance, sentinel surveillance and EHRs.
2. In-depth analysis of the existing MS system, in order to understand operating process and installed software. This was accomplished through several meetings with the National coordinator of MS, who was also a supervisor of this dissertation, and other members of MS staff.
3. Conceptualization of Sentinela+ architecture for data collection from EHRs and reporting to MS.
4. Development of Sentinela+ client application to fill MS notification forms. This activity included the following steps:
   (a) Design of a paper prototype of the application, which was submitted to approval of the MS coordinator.
   (b) Development of the application employing agile software development methods: the application modules were validated by the MS coordinator as they were being completed and revised according to the feedback received.
5. Development of a MedicineOne connector in the client application, to handle data extraction from this EHR system.
6. Development of a MedicineOne-Sentinela Web service, to extract data from MedicineOne database and return it to the client.
7. Development of Sentinela+ server for data storage.
8. Development of a web client to extract datasets from the Sentinela+ server in the format that could be imported to INSA server’s database.
9. User validation of the Sentinela+ system. Active sentinel physicians in MS tested a fully functional prototype of the system and evaluated through a user’s satisfaction questionnaire, derived from feedback.
from the Questionnaire for User Interaction Satisfaction (QUIS).

10. Evaluation of the system’s performance, through measurements of response times and effect on the EHR system performance.

1.5 Dissertation Outline

This dissertation comprises five more chapters besides the current chapter, for a total of six chapters.

- Chapter 2 provides a state of the art to the project, presenting the findings of the literature study.
- Chapter 3 includes a detailed analysis of current MS network.
- Chapter 4 introduces Sentinela+, detailing its architecture, development and implementation processes, and functionalities.
- Chapter 5 discusses the evaluation of Sentinela+.
- Chapter 6 concludes this work, evaluating the whole approach and debating whether its goal were achieved, and introduces directions for future work.
2 State of the Art

Contents

2.1 Public Health .......................................................... 8
2.2 Public Health Surveillance .......................................... 9
2.3 Sentinel Surveillance .................................................. 9
2.4 Electronic Health Record ............................................ 14
2.5 Summary ................................................................... 17
2.1 Public Health

Public health refers to a set of organized measures to prevent disease, promote health and well-being and prolong life, at a population level [1]. As the definition suggests, it focus on prevention and entire populations, rather than on treatment and individual patients. Public health is primarily a governmental activity.

Public health has three core functions: assessment, policy development and assurance. Assessment involves monitoring the health status of populations, identifying health problems and defining health priorities. Policy development concerns the establishment of public policies to solve health problems and priorities identified during assessment activities, in order to improve the health status of populations. Finally, assurance refers to the responsibility public health agencies have towards the populations: guaranteeing every member has access to necessary health promotion and disease prevention services [2].

The three core functions comprehend a set of ten essential public health services, which are a guiding framework for the tasks public health systems must carry out to achieve their mission [3]. The ten essential services are:

1. Monitoring health status in a community.
2. Diagnosing and investigating health problems in the community.
3. Informing, educating and empowering the community regarding these health problems.
4. Mobilizing community partnerships to identify and solve the health problems.
5. Developing policies to support individual and community efforts to improve health.
6. Enforcing laws and regulations that protect public health and ensure safety.
7. Linking individuals who require personal health services to the appropriate providers.
8. Assuring a competent public and personal health care workforce.
9. Evaluating personal and population health services.
10. Researching new insights and innovative solutions to health problems.

The essential services falls under one of the three core functions, as illustrated in Figure 2.1

![Figure 2.1: Ten essential services of public health. Each service is encompassed within one of the core functions (assessment, policy development, assurance) except for research, which is transversal to the other services. Image from http://www.health.state.mn.us/divs/opi/gov/chsadmin/intro.html](http://www.health.state.mn.us/divs/opi/gov/chsadmin/intro.html)
2.2 Public Health Surveillance

In public health, assessment activities rely heavily on surveillance. Public health surveillance is the continuous, systematic collection, analysis and interpretation of health-related data by public health agencies, and also the dissemination of these data to entities responsible for preventing and controlling disease and promoting healthy habits [4]. It provides a factual basis that can be used by public health agencies to set priorities, establish programs and intervene.

The value of a surveillance system is measured by its efficiency and effectiveness in collecting, analysing and delivering data [5]. Surveillance data allows to determine the necessity for public health interventions and provides feedback on the impact of those interventions. If timely and complete, it grant useful evidence to decision makers, empowering them to choose the best course of action [6].

Public health requires a wide variety of interventions, ranging from quick (e.g. preventing the spreading of acute infectious diseases) to prolonged (e.g. study of mortality causes) actions. Differences in the type and frequency of information and action needs lead to different types of surveillance. There are two main categories of surveillance: passive and active.

Passive surveillance consist of routine reporting of disease cases by physicians, laboratories and other health care professionals. Public health agencies just wait for the reports to come to them and do not stimulate reporting directly, hence the "passive" designation. Reports include routinely collected data such as hospital discharge summaries, mortality data and billing data, and reports of notifiable diseases (conditions that health care professionals must report by law to public health authorities) [7]. Data required from health care professionals is minimal, but there are few incentives to report and they often do not realize the importance of the information provided, which can easily translate into under-reporting. Overall, passive surveillance it is inexpensive and critical to monitor populations health status, but since it has numerous data providers it is difficult to guarantee timely, quality data [6].

In active surveillance, public health staff members contact physicians, hospitals, laboratories, schools, amongst others, to obtain information regarding certain health events [6]. It is frequently seasonal and coincident with periods of high disease frequency, since normally allows to detect a higher percentage of actual cases than passive surveillance [7]. It provides accurate and timely information, but requires more time and resources.

When it is not possible to obtain data regarding the entire population, sentinel surveillance is often applied. Sentinel surveillance concepts were particularly relevant to this project, and are described in greater detail in Section 2.3.

2.3 Sentinel Surveillance

2.3.1 Definition and Historical Perspective

Sentinel surveillance is a type of public health surveillance, in which a restrict number of selected healthcare professionals or sites voluntarily provides regular and standardized reports on a set of
defined health conditions, generally using a specific reporting tool (e.g. paper forms, web applications) [8]. "Sentinel" designation is used since it comprises a group of people keeping watch for particular diseases of interest.

Sentinel surveillance systems can supply a sample of information to assist public health decision making, when data are not available on the whole population. If the sample of healthcare professionals or sites is carefully selected, data collected in such systems can be used to indicate trends in the entire target population, identify disease outbreaks and study disease incidence in communities. It is an effective, rapid and flexible method for detecting large public health situations, requiring few resources. Since it involves a limited network of voluntary reporting healthcare professionals it can be used to collect high quality, high specificity data that would not be possible to collect for the entire population. However, since it uses a limited number of reporting sites, its efficiency is largely decreased for detecting health events that are rare or beyond the reach of the reporting entities [6–8].

Sentinel surveillance dates from 1951, when a pilot study was carried out in the United Kingdom, with a small number of GPs registering data regarding mortality in their patients [9]. In 1969, the Royal College of General Practitioners developed the Weekly Returns Service (WRS), in which a group of GPs summarized diagnoses and consultations, providing weekly reports regarding specific health events and a defined population. This new reporting method was very simple and effective [10]. Similar systems based on this approach were created afterwards in several European countries, such as Netherlands in 1970, Belgium in 1979, France in 1984, Switzerland in 1986, Spain and Portugal in 1989. Aforementioned systems also comprised a group of GPs who collected data about disease indicators and submitted weekly reports regarding these indicators [11]. Over the years, multiple works carried out by these sentinel systems established sentinel surveillance’s validity and consistency [9]. Currently, sentinel surveillance systems are used in most European countries and in developed countries in general. Sentinel surveillance importance is widely acknowledged, particularly when it comes to influenza surveillance.

2.3.2 Importance on Influenza Surveillance

Influenza is an acute viral infection that spreads around the world in a yearly outbreak. It is a serious public health problem that causes severe illness (3-5 million cases annually), excess mortality (250,000 to 500,000 deaths annually), burdens health services and affects the economy, through lost workforce productivity [12].

Vaccination is the most effective way to prevent infection. Since influenza viruses undergo constant antigenic change, it is necessary to ascertain what influenza viruses are circulating and detect new variants of the virus, to determine which influenza vaccine components must be selected each year. Furthermore, it is also required to assess geographical and temporal distribution of influenza activity and measure impact of influenza on hospitalizations and deaths [13]. Thus, both virological and epidemiological surveillance of influenza are needed. Sentinel surveillance plays a critical role in epidemiologic surveillance of influenza, recurring to clusters of sentinel networks that cover large areas, such as the European Influenza Surveillance Network (EISN) and the U.S. Outpatient Influenza-like
Illness Surveillance Network (ILINet).

2.3.2.A European Influenza Surveillance Network (EISN)

The EISN, coordinated by European Centre for Disease Prevention and Control, performs both epidemiological and virological surveillance of influenza. Its goal is to provide public health professionals and decision makers in European Union (EU) and European Economic Area necessary information to assess influenza activity in Europe and take appropriate action to reduce the impact of this disease [14].

The epidemiologic surveillance of influenza in the EISN is based on sentinel surveillance reports from multiple systems throughout Europe. Sentinel surveillance systems comprise a network of GPs who report the weekly number of patients seen with ILI or Acute Respiratory Infection (ARI), or both, to the national focal point for influenza surveillance. Some systems also include paediatricians (e.g., France, Germany, Spain) and physicians with other specialisations (e.g., Lithuania, Switzerland). The physicians in sentinel networks cover 1-5% of the population of respective country, guaranteeing statistical significance of the sample. Additionally, some sentinel physicians also extract biological samples from patients seen with ILI and ARI, which are tested for influenza virus and several other respiratory virus. Some countries also conduct sentinel surveillance for hospitalized cases of ILI and severe acute respiratory infection [14].

Furthermore, weekly surveillance reports containing a clinical and virological overview of the current week and of the season are published using data collected in the EISN. These data are made available online at the Flu News Europe web site [1]. Figure 2.2 shows an example of the type of information that can be found in this site, regarding influenza primary care data obtained through sentinel surveillance in EISN.

![Figure 2.2: Example of a plot using data collected through sentinel surveillance in the EISN. It represents the number of influenza viruses detections by subtype from week 40 of 2014 to week 17 of 2015. Image from https://www.flunewseurope.org/](https://www.flunewseurope.org/)
2.3.2.B U.S. Outpatient Influenza-like Illness Surveillance Network (ILINet)

In the United States, influenza surveillance is coordinated by Centers for Disease Control and Prevention (CDC), which collects, compiles and analyses data on influenza activity year round. Influenza surveillance is organised into five categories: virological surveillance, outpatient illness surveillance, mortality surveillance, hospitalization surveillance and summary of the geographic spread of influenza [15].

Outpatient illness surveillance is achieved through a sentinel surveillance system called ILINet. This nationwide network comprises approximately 2900 healthcare professionals, divided into smaller networks spread throughout 50 states, and operated by state-level health institutions. Sentinel healthcare professionals submit weekly reports, consisting of the number of patients diagnosed with ILI during a week grouped by five age groups, and the total number of patients seen by the physician during that week. Reports are submitted to CDC by the internet, phone or fax [15].

Data reported by sentinels are combined with data from other influenza surveillance sources, providing a picture of influenza virus in the entire country. CDC publishes a weekly surveillance report, called FluView. Reports are made available online at the CDC website comprising data summaries and plots discriminated by surveillance source. Figure 2.3 shows demonstrative examples of the plots obtained using influenza data collected through sentinel surveillance systems in ILINet.

![Figure 2.3: Examples of plots using data collected through sentinel surveillance in ILINet. Weekly consultation rates for influenza-like illness in the U.S. in the current year (red line), comparing it to the weekly consultation rates in previous years. Images from http://www.cdc.gov/flu/weekly/](http://www.cdc.gov/flu/weekly/)

*There was no week 53 in the previous influenza seasons displayed above, therefore the week 53 data point for those seasons is an average of weeks 52 and 1.*

*http://www.cdc.gov/flu/weekly/*
2.3.3 Sentinel Surveillance Information Systems

One of the main concerns in sentinel surveillance is its information systems infrastructure. In order to be effective, sentinel surveillance requires timely and quality data. However, since it relies on voluntary notifications from a group of health care professionals, it is important to ensure data collection processes are easy and quick. Sentinel surveillance information systems have evolved, adopting different approaches and methodologies, with particular focus on enhancing data collection processes.

As mentioned, most sentinel surveillance systems are based on networks of GPs that report data on a set of clinical events weekly. Initially, physician's reports consisted of filling out paper forms, sent by fax or post mail to public health agencies, where data was entered into computers and stored in local database, for further processing and analysis.

One notable exception was the French sentinel network, Réseau Sentinelles, which favoured the use of computer systems to report events since its creation. In 1996 it adopted a visionary reporting tool: a web based interface, used to transmit data through secure internet connections. This tool is still used presently. Data regarding cases of notifiable health events are compiled on a large MySQL database. These data are used mainly to calculate incidence rates and trace temporal and geographic evolution of the clinical events. Additionally, this system encompasses a web platform for information retrieval, accessible by any person. It allows to explore the database through an user-friendly interface, presenting data as maps, time-series and tables [16]. Since then, other systems adopted web based interfaces as the preferred reporting tool. However, many sentinel networks still resort exclusively to paper forms, or use both Web and paper for the notification process.

In 2012, the German sentinel network, Arbeitsgemeinschaft Influenza, introduced a new reporting tool: SEEDARE. This tool can query certain EHR systems and find cases of ARI that need to be reported. Computer-aided detection is based on diagnosis codes. Moreover, additional data regarding ARI occurrences necessary for sentinel reports are also extracted, including: age, gender, date of consultation, information about an eventual sick leave/hospitalization and influenza vaccination for each patient with ARI. Afterwards, physicians generate a file containing all these data, which are sent by e-mail weekly [17]. This system provides an opportunity to diminish the workload for physicians and also to ensure timeliness and stability of collected data [18].

In developing countries national population-based surveillance systems are harder to establish. Thus, sentinel surveillance provides an effective alternative to monitor certain health problems. For instance, in Africa, sentinel surveillance systems are being implemented to monitor severe epidemics, such as malaria. Due to the nature of these countries infrastructures, information systems differ from the ones found in developed countries.

In Madagascar a sentinel syndromic-based surveillance system was set up in 2007, by the Madagascar Ministry of Health and the Institute Pasteur in Madagascar. Initially, it focused on surveillance of fever symptom. Three diseases associated with fever were also selected for surveillance: malaria confirmed cases (positive result in rapid diagnostic test), influenza-like illness (fever with cough and sore throat) and arbovirus (fever without respiratory symptoms and the combination of at least two
of the following symptoms: headache, arthalgia, myalgia-like backache, skin rash, retro-orbital pain, haemorrhagic manifestations). Later, diarrheal disease was also included. Surveillance was based on data collected by GPs, including gender, age, consultation date and time and symptoms manifested of each new patient. Reports consisted on paper forms submitted weekly. Additionally, during working days GPs also reported data daily via encrypted short message service (SMS), regarding the number of fever, rapid test confirmed malaria, ILI, arboviral and diarrheal disease cases and the total number of consultations on each day. By using this reporting method, it was possible to detect peaks in fever occurrences and determine which disease was the most probable cause (arbovirosis, influenza, malaria) on a daily basis. Peak occurrences were reported immediately to regional health officials and public health officials at the Ministry of Health [19].

This is the first description of a sentinel surveillance system based on daily SMS reporting. It demonstrated the possibility of using this method as a simple and low-cost (costed less than 1$ per month) that may be applied in developing countries. Later, in 2010, a similar sentinel surveillance system based on SMS reporting was also successfully implemented in Ethiopia [20].

2.4 Electronic Health Record

An EHR is a repository of electronically maintained information about an individual’s health status and health care, designed to serve multiple uses and users [21]. It includes a wide range of heterogeneous data such as consultations registry, problems, medication, allergies, laboratory tests results, radiology images, vital signs, demographics and billing information.

EHRs can easily overcome some constrains of paper records. An EHR can display data in several ways, incorporate multimedia information (e.g. radiology images) and its content is more organized and legible. Furthermore, EHRs can be accessed from multiple distributed sites at the same time, unlike paper records which have a single physical copy. However, an EHR is not merely an electronic version of a medical paper record. It is much more comprehensive, including tools that analyse data inputs in order to assist providers in delivering higher quality care to patients [21].

One of these tools is Clinical Decision Support (CDS). CDS provides clinicians, staff, patients or other individuals with knowledge and person-specific information, intelligently filtered or presented at appropriate times, to enhance health care [22]. In other words, CDS tools take into account each patient’s clinical context to provide valuable information to decision makers. Notice that they do not perform decision making: the ultimate decision always rests with human decision makers. Some examples of features in CDS systems include computerized alerts and reminders (e.g. patient’s allergy to certain medication), diagnostic support tools, patient data summaries and display of clinical guidelines. Other feature EHRs include is Computerized Physician’s Order Entry (CPOE). A CPOE system is a tool that enables physicians to enter medical orders into a computer system, replacing traditional paper medical orders such as prescriptions and requests for laboratory and imaging tests [23]. Medical orders performed through CPOE systems are complete and clear, which is not always the case in paper orders, often incomplete or illegible. Studies suggest that serious medication errors
can be reduced by 83% when using a CPOE system coupled with a CDS system that emits alerts regarding physicians’ prescriptions [24].

Furthermore, patients visit different locations for different types of care (primary care centres, hospitals), their data end up scattered in several different information systems. Thus EHR systems must ensure clinical information exchange across different information systems in an appropriate and secure manner. To facilitate data exchange, EHR systems must store data in a structured way. Highly structured data includes patient identification numbers, names, dates, diagnose and medication codes. In order to collect such data EHRs typically ask users to enter information through structured forms, with fields that comprise drop-down lists or restrict data entry to controlled vocabularies. These data are easier to collect and exchange between systems because they are pre-defined, computer-readable and easily accessible from a database. Highly structured data can easily be aggregated, reported and transmitted to other systems for multiple purposes. However, physicians often consider that excessively structured data is a constrain to medical practice, impeding them to document a patient’s clinical history in a manner they find the most effective. Thus, most EHR systems often allow flexible documentation through narrative free-text, which is irregular, ambiguous and more difficult to exchange between systems [21] [25].

As described so far, comprehensive EHRs lead to clear improvements in clinical practice, enhancing the quality of care and reducing medical errors. Additionally, there are other advantages of using EHRs at an organizational level, such as increased revenue (decrease of billing errors, lost charges) and averted costs (reduced staff for patient management, increased utilization of tests). Furthermore, EHRs can improve the ability to conduct research and the overall population health in the long term, adding value at a societal level. However, EHR systems have some drawbacks. Implementing such systems implies financial costs derived from purchasing and installing software and hardware. Additionally, there are also maintenance costs, since hardware must be replaced and software updated regularly. Adopting EHRs may temporarily disrupt work-flows for healthcare professionals, since they have to learn how to operate the new system. One study estimated a productivity loss of 20% in the first month, 10% in the second month, and 5% in the third month, with productivity subsequently returning to its original levels. Other studies estimated that EHR users spent 134.2 hours learning the new system [24].

When balancing the advantages and disadvantages of these systems, they are beneficial, especially at the society level. EHRs are increasingly becoming the standard of care for office-based medical practices.

### 2.4.1 EHR usage in Europe

A study conducted in 2007, ordered by the European Commission [26], provided some relevant information regarding the use of information and communication technologies amongst GPs exercising their professional activity in the EU member states, particularly concerning the use of the EHR.

The study refers that computers and internet connections are available in most of the GP practices in Europe, with the electronic storage of administrative and clinical patient data gradually becoming a
norm. Data regarding diagnostics and medication prescriptions are the most frequently stored (92%), followed by data on basic medical parameters *e.g.* allergies (85%), laboratory results (81%), symptomatology and motives for consultation (79%), treatment outcomes (67%) and radiological images (35%). 80% of administrative patient data are stored electronically in GP practices. Furthermore, 76% of the member states store health data in a structured format, which facilitates the automatic processing of the data in other electronic systems.

However, data exchange between EHR systems was yet low. Clinical data transmission to other health care providers is not common (10%). 15% and 10% of administrative data are transferred for reimbursements and to other health care providers, respectively. Data exchange between laboratories and clinical care is the only exception (40%).

Notice that these numbers correspond to the year 2007; they indicate trends regarding the use of EHR in EU, but they are likely to have changed presently.

### 2.4.2 EHR in the NHS

Information system infrastructure in the National Health System comprises a set of informatics systems developed by Associação Central de Sistemas de Saúde (ACSS), which coexist in each health institution with a set of complementary systems, acquired in the market or developed in-house [27]. EHR usage is a common practice, with two major EHR systems currently used in the NHS: Sistema de Apoio ao Médico (SAM) and MedicineOne.

#### 2.4.2.A SAM

SAM derived from SONHO and SINUS, coordinated by the ACSS. SONHO is an administrative management system used to handle patients data, used in the majority of the NHS hospitals. SINUS is a similar system but applied to primary care context [28].

The necessity to create an EHR system led the developers of SONHO and SINUS to implement a medical module: SAM. SAM is an application used to collect and organize the information processed by physicians during medical practice. There are two versions of SAM: one for primary care and one for hospital care. In essence, both share the same functionalities, but comprise specific features adequate to each environment (hospital and primary care) [29].

This system's technology is based on Oracle Forms 6 and Oracle Reports software, that interacts with a local Oracle database, where clinical data is stored. Modules that do not concern clinical data are accessed through interaction with SONHO or SINUS. SAM is used by approximately 90% of the Portuguese health care centres [30].

#### 2.4.2.B MedicineOne

MedicineOne is a software for management of patients’ clinical and administrative information. It was created in 1989 for primary care services and since 2006 expanded its features to cover hospital needs. Unlike SAM, MedicineOne provides specialized modules for physicians, nurses and administrators in a single application. This system resorts to Microsoft technology written in C# and uses
2.5 Summary

Sentinel surveillance systems are an effective public health surveillance strategy, applied when it is not possible to obtain data on entire populations or more detailed variables about certain health condition are required. Currently, sentinel surveillance is used all over the world, with particular importance on influenza surveillance.

Information systems in sentinel surveillance must have a special concern regarding data collection processes. The use of paper forms is falling into disuse, and web based interfaces that can transmit data through secure network connections are currently the state-of-the-art. However, the increasing use of the EHR presents an opportunity to collect data from clinical care and use it for sentinel surveillance purposes. The German sentinel network recently introduced a tool that extracts cases on a notifiable health condition from several EHR systems, with detection mechanisms based on diagnoses codes. Additionally, this tool collects administrative and vaccination data on the diagnosed patients. It is possible to go further, and extract even more data, such as clinical data and medication prescription data, which are registered in virtually every EHR system.

The move from manual reporting to automated data collection from various data sources is becoming a reality, and public health agencies must take advantage of the benefits of electronic reporting. Interoperability between sentinel surveillance information systems and EHR systems to detect cases of notifiable health events, associated with the possibility of extracting further data regarding these events, can deliver a great portion of sentinel surveillance data needs, and facilitate data collection processes.
Médicos Sentinela

Contents

3.1 Overview ................................................................. 20
3.2 Notifiable Health Events ................................................. 22
3.3 Notification Forms ...................................................... 22
3.4 Notification Tools ...................................................... 26
3.5 Data Retrieval .......................................................... 28
3.6 Summary .................................................................. 29
The information presented throughout this chapter can be consulted in: [32] [33] and [34]. Furthermore, personal contact with public health professionals that coordinate MS network and MS sentinel physicians provided additional information that was used in this chapter. No further references will be made along the chapter.

3.1 Overview

Médicos Sentinela (MS) is a Portuguese sentinel surveillance system. It was developed in 1989 and it is coordinated by the Epidemiology Department at National Health Institute Dr. Ricardo Jorge.

Being a sentinel surveillance system, MS includes a network of GPs, who collect data regarding a set of clinical events relevant to the public health domain. Collected data are entered into notification forms, that can either be web forms or paper forms. These data are used by public health professionals, for the following purposes:

- Contribute to epidemiological surveillance in Portugal, through the estimation of the incidence rate of certain diseases and other clinical events relevant to public health.
- Identify outbreaks early.
- Contribute to epidemiological research, specially regarding primary care, through the creation of national database.

As mentioned, a key element of MS surveillance system is its the network of GPs, who collect data regarding a set of pre-defined clinical events. Currently there are 107 sentinel GPs in the network, of which 66 (61.7%) are active, i.e. have notified at least once during the ongoing year. Sentinel physicians’ geographic distribution is shown in figure 3.1.

Figure 3.1: Geographical distribution of sentinel physicians in MS network: number of sentinels per district and in the autonomous regions.
Participation in MS network is voluntary, and in order to take part, a physician must exercise his professional activity in primary care centres encompassed in the NHS. The reason why the network includes exclusively primary care GPs relies on two factors: primary health care data is more comprehensive and in-depth than hospital care data and in primary care, a list of patients is assigned to each physician, which means the set of patients to whom each a physician provides care is well-defined.

The later is relevant since it makes possible to define precisely the maximum population under observation (MPUO) covered by sentinel physicians. Maximum population under observation corresponds to the sum of all patients in the lists of patients of all sentinels. It is calculated using the following equation:

\[ MPUO = \sum_{m=1}^{M} N_m \]  

Where:

- \( N_m \) Represents the number of patients belonging to the list of patients of physician \( m \).
- \( \sum_{m=1}^{M} \) Represents the sum of all physicians in MS network.

However, only a portion of the total number of physicians actually reports actively during certain periods of time, which affects the effective population under observation (EPUO). The EPUO is calculated for a certain period of time and depends upon the number of active physicians during that period. EPUO for week \( t \) is given by:

\[ EPUO_t = \sum_{m=1}^{M} N_m I_{tm} \]  

Where:

- \( N_m \) Represents the number of patients belonging to the list of patients of physician \( m \).
- \( \sum_{m=1}^{M} \) Represents the sum of all physicians in MS network.
- \( I_{tm} \) Variable that equals 1 if physician \( m \) was active during week \( t \) and 0 otherwise.

Yearly EPUO is computed as the average of weekly effective population under observation, throughout the 52 weeks of the year:

\[ EPUO_{year} = \sum_t \frac{EPUO_t}{52} \]  

Results obtained using data regarding the effective population under observation are generalized to the Portuguese population. In other words, the population under observation serves as a sample of the whole population. In 2014, MS covered an effective population of 33,656 patients, which corresponds to approximately 0.32% of the Portuguese population.

Moreover, since the population under observation depends upon the number of physicians in MS network, as can be noted in equations 3.1 and 3.2, it is desirable to have as much physicians in the
network as possible, to increase the population covered and its statistical representativeness as a sample of the whole population.

In order to determine the number of patients covered by each physician, sentinels must submit an annual list with the number of patients they provide care to, distributed per gender and age group. Discrimination per gender and age group makes possible to define subsets within the population under observation as well.

A network with the structure described has several advantages:

- Possibility of collecting extra variables, in addition to data recorded in health records, inherent to sentinel surveillance approach.
- Well-defined population under observation, since the network comprises only primary care physicians with well-defined lists of patients.
- Easy data validation, due to a relatively small number of participant physicians.

### 3.2 Notifiable Health Events

MS focuses on the surveillance of a limited number of health events, which are determined by public health professionals at INSA in the beginning of each civil year. These events are referred to as notifiable health events.

For a health event to be subject of surveillance in MS, it must meet certain criteria. First and foremost, it must be of national interest. Thus, events under observation each year tend to be the most significant for public health surveillance. Since physicians who collect data are exclusively primary care physicians, health events under surveillance must also be of usual incidence in primary care, to allow continuous data collection. Additionally, to increase quality and frequency of reports, it is important to choose events that are easy to report and of interest to physicians.

All these aspects translate into some health events being under surveillance for several years whereas others are only reported for few years or an isolated year. Presently there are six notifiable health events: influenza-like illness, hypertension, diabetes mellitus, cerebrovascular accident (CVA), acute myocardial infarction (AMI) and oral anticoagulants prescription.

### 3.3 Notification Forms

A notification form is a set of fields regarding the six notifiable health events described in Section 3.2, to be filled with both clinical data about these events and personal data concerning the patient who manifests them. Sentinel physicians must fill a notification form, if a case arises during medical practice and meets the following criteria:

- ILI case in which symptoms started in the current year.
- AMI, CVA, diabetes or hypertension case in which the respective disease was diagnosed in the current year.
• Oral anticoagulants prescription case in which this type of treatment was prescribed for the first time, i.e. if the patient never took anticoagulants before.

Additionally, sentinel physicians notify exclusively cases verified in patients belonging to their list of patients. If they provide care to a patient that does not belong to their list, who shows up with a case of a notifiable event, they shall not fill a notification form. Inversely, if one of their patients manifests a notifiable event, but it is detected by another physician, sentinel physicians must fill a notification form when they are aware of the situation. This situation is fairly common, since for some of the clinical events, such as CVA or AMI, patients are frequently diagnosed and treated in hospitals, and when discharged continue monitoring and treatment of the disease in primary care.

In order to better understand notification forms, it is necessary to take a closer look at their fields. These fields can be divided into general fields, which are common to every notifiable event and specific fields, which differ for each event. For each notification form physicians must fill the general fields, regardless of the event, and the specific fields concerning only the event being reported. There is a third subset of fields, described in Section 3.3.7 which is only filled in a peculiar situation

### 3.3.1 General fields

There are five general fields, presented in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification key</td>
<td>Unique identifier for each notification form.</td>
</tr>
<tr>
<td>Notification date</td>
<td>Date when the notification form was filled.</td>
</tr>
<tr>
<td>Health event identification</td>
<td>Identification of the notifiable health event corresponding to case reported in the notification form.</td>
</tr>
<tr>
<td>Date of occurrence</td>
<td>Date when the case was verified.</td>
</tr>
<tr>
<td>Patient date of birth</td>
<td>Date of birth of the patient who manifested the event.</td>
</tr>
<tr>
<td>Patient gender</td>
<td>Gender of the patient who manifested the event.</td>
</tr>
</tbody>
</table>

Table 3.1: General fields in notification forms. The first column shows the fields and the second column describes the required data input for each of these fields.

The notification key uniquely identify each form. It includes the identification number of the sentinel that filled the form, the notification tool used, the year and the sequential notification number for that sentinel in that year, in the format sentinelnumber_toolyear_notificationnumber. Notice that cases date of occurrence vary in agreement to health event that is being notified.

• Influenza-like illness: start date of the symptoms.
• Hypertension, diabetes, CVA, AMI: diagnosis date.
• Oral anticoagulants prescription: prescription date.
3.3.2 Influenza-like Illness Specific Fields

For influenza-like illness events, there are fifteen specific fields, described in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sudden onset, Fever, Weakness/Prostration, Headache, Myalgia, Cough, Sore throat and inflammation of the nasopharyngeal mucosa and no significant respiratory signs, Difficulty breathing, Chills</td>
<td>“Yes”, if the patient manifested any of the symptoms described in the fields. “No” if the patient did not manifested any of the symptoms described in the fields. “Undetermined” if it was not clear if the patient manifested any of the symptoms described in the fields.</td>
</tr>
<tr>
<td>Did the patient had contact with another flu patient? Was the patient vaccinated against seasonal flu? Will an antiviral be prescribed?</td>
<td>Data input must answer the question put in the fields with one of three options: “Yes”, “No” or “Undetermined”.</td>
</tr>
<tr>
<td>Does the patient has a chronic disease? Which one?</td>
<td>Data input must answer the question put in the fields with one of three options: “Yes”, “No” or “Undetermined”. Additionally, if the answer is “Yes”, the physician must specify which disease.</td>
</tr>
<tr>
<td>Is the patient pregnant?</td>
<td>Data input must answer the question put in the fields with one of three options: “Yes”, “No” or “Not Applicable”.</td>
</tr>
<tr>
<td>Was nasopharyngeal, exudate sent to laboratory?</td>
<td>Data input must answer the question put in the fields with one of two options: “Yes” or “No”.</td>
</tr>
</tbody>
</table>

Table 3.2: Influenza-like illness fields in notification forms. The first column shows the fields and the second column describes the required data input for each of these fields.

3.3.3 Oral Anticoagulants Prescription Specific Fields

For oral anticoagulants prescription events, there are three specific fields, described in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prescribed anticoagulant</td>
<td>Active ingredient of the prescribed anticoagulant.</td>
</tr>
<tr>
<td>Clinical problem that led to prescription</td>
<td>Medical term designating the clinical problem that led to the prescription of the anticoagulant.</td>
</tr>
<tr>
<td>Who prescribed the medication?</td>
<td>Identification of the physician who prescribed the anticoagulant: “Me”, if the physician who is filling the notification form prescribed the anticoagulant, or “Other”, if it was a different physician. If the option “Other” was chosen in the previous field, the physician notifying must specify the name.</td>
</tr>
</tbody>
</table>

Table 3.3: Oral anticoagulants fields in notification forms. The first column shows the fields and the second column describes the required data input for each of these fields.
3.3.4 Diabetes Specific Fields

For diabetes mellitus events, there are four specific fields, described in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of diabetes</td>
<td>Identification of the type of diabetes: type 1, type 2, gestational, other or unknown.</td>
</tr>
<tr>
<td>Initial treatment (diabetes type 2)</td>
<td>Initial treatment recommended for the disease, comprising the active ingredient of up to three prescribed medications and one non-pharmacological measure indicated.</td>
</tr>
<tr>
<td>Who prescribed the medication? (diabetes type 2)</td>
<td>Identification of the physician who prescribed the initial treatment referred in the previous field: &quot;Me&quot;, if the physician who is filling the notification form prescribed the treatment, or &quot;Other&quot;, if it was a different physician. If the option &quot;Other&quot; was chosen, the physician notifying must specify the name.</td>
</tr>
<tr>
<td>Was initial treatment maintained?</td>
<td>Data input must answer the question with one of three options: &quot;Yes&quot;, if the initial treatment (referring to the previous field) was attained, or &quot;No&quot;, if the initial treatment was not maintained. This question only applies if the previous field was filled with the option &quot;Other&quot;.</td>
</tr>
</tbody>
</table>

Table 3.4: Diabetes fields in notification forms. The first column shows the fields and the second column describes the required data input for each of these fields.

3.3.5 Hypertension Specific Fields

For hypertension events, there are three specific fields, described in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial treatment</td>
<td>Initial treatment recommended for the disease, comprising the active ingredient of up to three prescribed medications and one non-pharmacological measure indicated.</td>
</tr>
<tr>
<td>Who prescribed the medication?</td>
<td>Identification of the physician who prescribed the initial treatment referred in the previous field: &quot;Me&quot;, if the physician who is filling the notification form prescribed the treatment, or &quot;Other&quot;, if it was a different physician. If the option &quot;Other&quot; was chosen, the physician notifying must specify the name.</td>
</tr>
<tr>
<td>Was initial treatment maintained? (diabetes type 2)</td>
<td>Data input must answer the question with one of three options: &quot;Yes&quot;, if the initial treatment (referring to the previous field) was attained, or &quot;No&quot;, if the initial treatment was not maintained. This question only applies if the previous field was filled with the option &quot;Other&quot;.</td>
</tr>
</tbody>
</table>

Table 3.5: Hypertension fields in notification forms. The first column shows the fields and the second column describes the required data input for each of these fields.
3.3.6 AMI and CVA Specific Fields

Regarding CVA and AMI, there are no specific fields. No further data is necessary, thus it is only necessary to fill general fields in notification forms, when these events occur.

3.3.7 No Cases and Absence Fields

When a physician does not verify any case of the notifiable health events during a certain period of time, he does not submit any notification form during that period. Additionally, if a physician is absent from medical practice, he does not submit notification forms as well. These situations have a different impact in the calculation of incidence rates: zero cases of certain health event means that its incidence rate is zero, whilst during physician's absence, no data is collected whatsoever, which does not necessarily means the incidence rate is zero. Evidently, it becomes necessary to make a distinction between these two situations. So, there are two additional fields in notification forms, to report no cases or absence, exclusively filled if no notification forms were submitted during the period of one week, to determine the motive behind it. These fields are presented in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Data input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of notifications period</td>
<td>Period of time during which the physician did not submit any notification form.</td>
</tr>
<tr>
<td>Motive for the lack of notifications</td>
<td>Reason for the absence of notifications described above, allowing to determine if the reason is no cases of clinical events or physician's absence (vacation, professional courses, illness, other).</td>
</tr>
</tbody>
</table>

Table 3.6: No cases and absence fields in notification forms. The first column shows the fields and the second column describes the required data input for each of these fields.

These fields are always left blank, except when reporting no cases or absence. Also, when reporting these situations, only these two fields are required and all the others should be left empty, with the exception of the notification key and filling date, which are always required.

3.4 Notification Tools

There are two tools to fill notification forms: a web application and paper. These tools are described in the following subsections.

3.4.1 Web Application

MS system has a web application designed to serve its needs. It was developed in an ASP.NET framework by an outsource IT services provider. It is incorporated in a web platform called RIOS, which contains a set of health information and surveillance web applications. Users can access RIOS platform through a web browser, where they can find MS web application, amongst other applications. This particular application will be referred to as MS-RIOS web application from now on.
Both sentinel physicians and public health professionals can access MS-RIOS with different features available depending on the type of user. There is an user authentication, ensuring only MS personnel can access it and also determining the access level (physician or public health professional). This chapter focus on the features used by physicians for reporting purposes. MS-RIOS features used by public health professionals are mainly used to return notification forms data, described later in Section 3.5.

Sentinels use MS-RIOS as a notification tool. The application comprises an ASP.NET web form interface, with Graphical User Interface (GUI) elements for data input such as text fields, radio buttons and drop-down lists, comprising all the fields described in Section 3.3. Physicians fill only the fields required for the case they are reporting. For instance, if reporting a case of hypertension, the physician fill only general fields (Figure 3.2) and hypertension specific fields (Figure 3.3).

When the form is completed, physicians submit it and its data are sent to a RIOS platform server. Moreover, MS-RIOS also allows physicians to search previously submitted notification forms. This is useful to keep track of reports, identify possible duplicated reports and validate submitted data, if requested by public health staff. Physicians can search by form unique key, filling date, clinical event, occurrence of an event date, patient’s age and gender.
3.4.2 Paper

In MS it is also possible to notify through paper forms. Paper forms also encompass all the fields described in Section 3.3. Physicians identify which clinical event they are reporting in the paper form and only fill the fields regarding that event, leaving fields concerning other events empty. Afterwards, paper forms have to be sent to the INSA via post mail, on a weekly basis. There, paper forms data are entered into MS-RIOS Web application, by MS system administrators. This reporting tool method is noticeably outdated, but it is continued since there are physicians in MS network who oppose or dislike the use of technology.

3.5 Data Retrieval

As mentioned, once notification forms are submitted through MS-RIOS Web application, their data are sent to RIOS server, running a Microsoft SQL Server Database Management System (DBMS). However, data have to be transferred to INSA servers, where they are managed in a dBase DBMS.

Data transfer between RIOS server and INSA server is accomplished through MS-RIOS Web application. MS system administrators can access features within the application that allow to export data from the RIOS server in DBF file format, typically used by dBase software systems. The files are exported into MS system administrators computers. Data can only be exported by notifiable health event. Consequently, each DBF file comprises data regarding cases of a single clinical event. Additionally, users also determine whether they want to export all data stored, all data since the last exportation or data submitted between certain notification dates or occurrence dates (Figure 3.4).

![Figure 3.4: Screenshot of MS-RIOS application, showing the interface that allows public health staff export data from the notification forms.](image)

MS system administrators also need to verify exported data. This task is easily performed, since DBF files can be opened in Microsoft Excel. If some data seems incorrect, system administrators contact the physician who filled the notification form in order to double check it. If necessary, the DBF file is edited in Excel, overriding the original file. Only then DBF files are ready to be imported into the final database, using a simple command in dBase's command window. This whole process is usually performed on a daily basis.

The whole process, since a sentinel physician detects a case of a notifiable event until data regarding that case are stored in INSA server is illustrated by Figure 3.5.
Figure 3.5: Notification process in MS network, from the moment a case of a notifiable event is verified by a sentinel, until its data are stored in INSA server, ready to be used for public health surveillance purposes.

Once data are stored in dBase, it is finally ready to be used for public health purposes, which comprise mainly the calculation of incidence rates. Incidence rates are directly calculated using dBase software. The incidence rate is the number of new cases per population observed in a given time period. Thus, a health event incidence rate is the quotient between the number of new cases of that event during certain period of time and the population under observation. The denominator utilized is the effective population under observation (see Section 3.1). Afterwards, an incidence rate is estimated for the Portuguese population.

For every notifiable health event, annual incidence rate and annual incidence rate per gender and age group are computed. Comparison between annual incidence rates over the years allows to monitor the evolution of these events in Portugal.

Additionally, for ILI the weekly incidence rate is also computed, in order to observe influenza evolution throughout the year. Weekly incidence rate is compared with a basal activity area, defined as the 95% upper confidence limit of an established baseline. This allows to identify disease outbreaks. There is epidemic activity whenever the incidence rate is above the basal activity area, with laboratory confirmation for influenza virus. Widespread epidemic activity occurs when the incidence rate is above the basal activity area for two or more consecutive weeks and manifests a tendency to increase, with laboratory confirmation for influenza virus.

Moreover, gathered data is also used for epidemiological research. Data collected in notification forms is detailed, and it goes behind collecting disease diagnosis. For instance, collected data regarding diabetes and hypertension can be used to describe the therapeutic profile for these diseases and collected data regarding anticoagulants prescription to determine which anticoagulants are more frequently prescribed, and which disease leads to this prescription.

3.6 Summary

Médicos Sentinela is a sentinel surveillance system, comprising a network of GPs who voluntarily notify cases of a set of health events. Notifications are performed through web and paper forms, with
There are some limitations inherent to MS network characteristics. Voluntary participation in MS network guarantees data quality and timeliness. However, it is also a downside, since the group of physicians that participate in MS network is a convenience sample of the GPs in the NHS, meaning the population under observation may not be representative of the Portuguese population. Thus, it is desirable to increase the number of sentinels in the network. Notice that increasing the number of physicians also increases the population under observation, and consequently, its representativeness of the Portuguese population.

Nonetheless, increasing the number of physicians is not an easy task. Data collection for MS surveillance requires time from the sentinel physicians. In a system based on data from patients who visit physicians, time efficiency is the key. The longer the reporting task takes, the more it disrupts the physician's workflow. If surveillance requires such an effort that it disturbs routine health care delivery, it leads to non-cooperation from physicians.

Additionally, it is necessary to stimulate a continuous active participation from the physicians within the network. Only 61.7% of the physicians who already constitute MS network actually notify actively, in the current year. And from that percentage, some do not notify continuously throughout the year. For instance, it is quite common for some physicians to only submit forms during influenza outbreaks.

This project seeks to enhance data collection processes in MS, by proposing a notification tool that uses EHR data to automated reporting, described in Chapter 4.
4
Sentinela+

Contents

4.1 Overview ................................................................. 32
4.2 Client Application ...................................................... 33
4.3 Sentinela+ Server ....................................................... 44
4.4 MedicineOne Connection ............................................. 46
4.1 Overview

4.1.1 Proposed Approach

During their professional activity, physicians already dedicate a significant amount of work time to data entry, primarily in the EHR. Studies show that physicians spend more time on data entry than on direct patient care [35]. EHRs focus primarily on supporting clinical activities, and are designed to provide patient-level data and provider-level decision support. However, they include comprehensive data that can be used for public health purposes. As discussed in the previous sections, notification forms in sentinel surveillance networks demand manual introduction of data regarding a set of health events under observation, usually on web or paper forms. For sentinel physicians, this represents an additional time consuming data entry process. Additionally, most of the data entered in notification forms is duplicated, since these data were already entered in the EHR, making the notification process a repetitive task. However, advances in information technology present an opportunity to integrate clinical systems and public health systems. Given that clinical systems capture a great deal of patients’ health data, it seems advantageous to use aggregates of data for public health purposes, such as automated reporting in sentinel surveillance.

The opportunity to study the use of EHR data for automated reporting in the Portuguese sentinel network Médicos Sentinela (MS) motivated this project. In MS, notifications are performed through web and paper forms as well. I proposed to develop and implement a system integrating MS and EHR systems. The basic concept is to detect all the cases of MS-notifiable health events from EHR systems, and extract data regarding these cases in order to automatically fill MS notification forms. Sentinel physicians manage the whole operation, by requesting extraction processes, validating extracted data and submitting these data for storage in servers. Figure 4.1 illustrates how the MS public health system could be integrated with clinical systems.

Figure 4.1: Comparison between the current system and the proposed systems. (Left) In the current system public health system and clinical system are disconnected. (Right) Integration between both systems can provide most of the data for automated completion of MS notification forms, with the physicians just over-viewing the notification process.

I developed and implemented the proposed system, called Sentinela+.
4.1.2 Architecture

Sentinela+ includes a client desktop application used to fill notification forms, with data extracted from clinical information systems. The client includes a module (connector) to communicate with clinical information systems. Additionally, Sentinela+ comprises a virtual server, to where notification forms data are sent. Datasets in the server can be exported through a web application, which can be accessed in a web browser by MS system administrators. These datasets are further processed and analysed by public health professionals. The system conceptualization is illustrated in Figure 4.2.

![Figure 4.2: Conceptualization of Sentinela+ system.](image)

4.1.3 Base Software

Sentinela+ client is a lightweight desktop application written in Java language. The GUI was developed in Netbeans IDE 8.0.1, using Java Swing components. Additionally, the client includes an embedded Apache Derby database, where data regarding notification forms are stored. Sentinela+ server runs Microsoft SQL Server 2008 as the DBMS. Users can return datasets from the Sentinela+ server in Comma Separated Values (CSV) format, through an ASP.NET web application, developed in Visual Studio 2013 and using code-behind the page in C# language, with an HTML user interface.

4.2 Client Application

A key element in Sentinela+ system is its client application, a tool used to fill MS notification forms. It consists of a desktop Java-coded application, rather than the state-of-the-art Web applications. The use of desktop applications for sentinel surveillance reporting was studied in the French sentinel network, revealing that desktop availability allowed to describe cases quickly and provided better integration in physicians work process [36]. Additionally, this method provides an alternative to the existing reporting tools in MS.

Sentinela+ client allows to request data extraction processes in the EHR system, receiving data and storing them on its local database. These data are used for automated filling of notification forms, displayed to users through the application GUI. Moreover, the client allows sending forms data to Sentinela+ server for further processing. Furthermore, users can use this tool to manually fill notification forms without extracting any data from the EHR system, as they do in the current reporting.
Sentinel+ client application comprises four key actions: CREATE, IMPORT, MANAGE and SUBMIT. These actions can be performed through the main window, which consists of a simple, user-friendly interface with one button for each action, as illustrated in Figure 4.3.

Each action button has different purpose:

- **CREATE** – Report a case manually.
- **IMPORT** – Request data extraction processes from the EHR, receiving the response and storing its data in the local database.
- **MANAGE** – Manage all extracted cases, that can either be displayed and completed, or deleted.
- **SUBMIT** – Send data to Sentinel+ server.

### 4.2.1 Notification Forms GUI

Notification forms data are displayed and edited through the application GUI. Sentinel+ forms consist of a set of Java Swing components displayed in a frame, including text fields, combo boxes, radio buttons, date choosers and push buttons. Moreover, the code behind each form manages the process of capturing data entered by users through GUI elements and process it for storage in the application's database.

Each health event under surveillance is notified through a different form (see Annex A). However, there is a general structure, common to all forms. As observed in Figure 4.4, for each clinical event, Sentinel+ forms comprise the event identification, event's general fields, event's specific fields and action buttons. The event identification and the event’s specific fields differ for each health event.
4.2.2 Data Model

The data model used to represent notification forms in Sentinela+ client is simple and straightforward: all notification forms data inputs, depicted in Section 3.3 are stored in the table NOTIFICATIONS, in the local client database. Additionally, this table comprises three additional data elements, which allow to manage the forms within the application:

- **internal_key** – Sequential number assigned to every notification form created, functioning as an unique identifier of the form within the application.
- **id_patient** – Identifies to which patient a notification form refers. This data element is removed before submitting the form data to Sentinela+ server, guaranteeing data are de-identified.
- **status** – Defines the form’s status: incomplete (0), complete (1), submitted (7) or deleted (9).

A listing of all fields in NOTIFICATIONS can be found in Table 4.1. Each column represents a different data input from notification forms. Thus, each row comprises all data regarding a single notification form. Data concerning cases extracted from the EHR are used to populate the table NOTIFICATIONS as much as possible, with each case data being stored in a single row. Additionally, data inputs performed through GUI elements are processed by the application’s code, and stored in NOTIFICATIONS. Data elements in the local database and GUI elements for data input are depicted in the following paragraphs.

Each notification form corresponds to a single notifiable health event. The identification of the event is set by default and stored in the field event_id, using the following correspondence: ILI – 1, diabetes – 2, hypertension – 3, AMI – 4, CVA – 5, Anticoagulants Prescription – 6. Additionally, every form is automatically assigned a sequential identification number, which is the primary key of
NOTIFICATIONS, stored in the field internal_key, and a status, stored in the field status. These two data elements are not visible to users but are necessary to identify and manage forms within the application.

Patient’s identification number is entered through a text field and stored in the field id_patient. Patient’s gender can be chosen using a combo box with two options: ‘M’ (male) or ‘F’ (female) and it is stored in the field gender_patient as an integer, using the following codification: male – 1, female – 2. Dates, including notification form filling date, the case date of occurrence and patient’s date of birth, are introduced into forms through date choosers and stored as dates in the fields filling_date, event_date and dob_patient, respectively.

Specific fields differ for each reportable clinical event. For ILI events, specific fields refer mostly to influenza symptomatology and other questions related to this disease. These fields are mostly questions that must be answered with yes, no, not applicable or undefined, captured through radio buttons with these options. Data inputs are stored in ILI fields in NOTIFICATIONS, using the following correspondence: ‘Yes’ – 1, ‘No’ – 2, ‘Not Applicable’ – 7, ‘Undefined’ – 9.

Diabetes, hypertension and anticoagulants prescription specific fields comprise similar fields between themselves. These fields include medication(s) prescribed, which can be selected from combo boxes comprising a list of active principles of medications prescribed for the aforementioned diseases. The respective medication’s ATC code is stored in NOTIFICATIONS, in fields dbt_treatment(1-3), hyp_treatment(1-3) and oacp_treatment. For diabetes and hypertension there is also a free text input field for non-pharmacological treatments, obtained through a text field and stored in the fields dbt_treatment4 and hyp_treatment4. Other fields include medication prescription initiative, which is collected through radio buttons with the options ‘Mine’ (user sentinel physician) or ‘Other’ (other physician). These data are stored in the fields dbt_initiative, hyp_initiative and oacp_initiative, using the codification ‘Mine’ – 1, ‘Other’ – 2. If the prescription initiative came from another physician, two additional fields must be filled in the form, regarding which physician prescribed the medication and if the sentinel physician changed that medication. The first field is captured through a text field and stored in dbt_initiativewho, hyp_initiativewho and oacp_initiativewho. The second field is only applicable for diabetes and hypertension; it is collected through radio buttons with the options ‘Yes’ or ‘No’, and stored in fields dbt_mantained and hyp_mantained, under the following codification: ‘Yes’ – 1, ‘No’ – 2. Furthermore, diabetes forms comprise a field to identify the type of diabetes, which can be filled through radio buttons with the diabetes types as options. Data are stored in dbt_type, using the codification Type 1 – 1, Type 2 – 2, Gestational – 3, Other – 4 and Unknown – 5. At last, anticoagulants prescription specific fields include a free text field to define the clinical situation that led to that prescription, which is stored in the field oacp_situation. AMI and CVA do not require any specific field, thus the forms do not comprise these section.
<table>
<thead>
<tr>
<th>Database field</th>
<th>Data type</th>
<th>Field description</th>
</tr>
</thead>
<tbody>
<tr>
<td>internal_key</td>
<td>Integer</td>
<td>Numeric key assigned to every notification form.</td>
</tr>
<tr>
<td>status</td>
<td>Integer</td>
<td>Status of the notification form.</td>
</tr>
<tr>
<td>id_patient</td>
<td>Integer</td>
<td>Identification number of the patient.</td>
</tr>
<tr>
<td>notification_key</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>notification_date</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>event_id</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>event_date</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>dob_patient</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>gender_patient</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_start</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_fever</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_weak</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_headache</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_myalgia</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_cough</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_inflammation</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_respiratory</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_chills</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_contact</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_vaccine</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_antiviral</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_chronic</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_chronicwhich</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>flu_pregnant</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>flu_exsudate</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>oacp_treatment</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>oacp_situation</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>oacp_initiative</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>oacp_initiativewho</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>dbt_type</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>dbt_treatment1</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>dbt_treatment2</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>dbt_treatment3</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>dbt_treatment4</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>dbt_initiative</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>dbt_initiativewho</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>dbt_maintained</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>hyp_treatment1</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>hyp_treatment2</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>hyp_treatment3</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>hyp_treatment4</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>hyp_initiative</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>hyp_initiativewho</td>
<td>Varchar</td>
<td></td>
</tr>
<tr>
<td>hyp_maintained</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>absence_week</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>absence_motive</td>
<td>Varchar</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General fields, described in Subsection 3.3.1.</td>
</tr>
<tr>
<td>IRI fields, described in Subsection 3.3.2</td>
</tr>
<tr>
<td>Anticoagulants prescription fields, described in Subsection 3.3.3</td>
</tr>
<tr>
<td>Diabetes fields, described in Subsection 3.3.4</td>
</tr>
<tr>
<td>Hypertension fields, described in Subsection 3.3.5</td>
</tr>
<tr>
<td>No cases and absence fields, described in Subsection 3.3.7</td>
</tr>
</tbody>
</table>

Table 4.1: NOTIFICATIONS table fields, data input types and description of each field.

Additionally, the client also stores data regarding notification activities performed by users, in table ACTIONS. This table comprises a single row record that is constantly changed according to user actions within the application, divided in five columns:

- **notification_count** – Number of notification forms submitted to Sentinela+ server.
• `last_import_date` and `last_import_time` – Date and time when the last data extraction from the EHR system was performed.

• `last_export_date` and `last_export_time` – Date and time when the last data sending to Sentinela+ server occurred.

The date and time of the last extraction process are used to determine which cases where previously extracted from the EHR. The number of notifications submitted to Sentinela+ server is necessary to determine each notification form key (see Section 3.3.1). The date and time of the last data sending are used to provide warnings, if users do not notify for a period longer than one week, as depicted later in this chapter.

### 4.2.3 Semi-Automated Notification Process

Semi-automated notification refers to the process of extracting the list of cases that needs to be notified and data regarding these cases from the EHR, so that these data can be used to automatically fill notification forms. Since not all data necessary to complete the forms can be extracted, sentinels have to manually enter any missing data before submitting the form, hence the semi-automated designation. This process can be managed from the desktop application.

Through the IMPORT action button, users call the Connector Module (see Figure 4.5), which allows to request an extraction process, importing data into Sentinela+ client and storing them in the local database. For each case extracted, a new row is added to `NOTIFICATIONS` table. The form is assigned an automatic key (`internal_key`) and the status 0, corresponding to a extracted form (`status`). The extracted case health event identification is detected and stored in the field `event_id`. Afterwards, extracted data elements for that case are stored in the respective event fields. The data row created represents an incomplete notification form, that users can either display and complete, or delete, using the MANAGE feature.

The Connector also determines the clinical system used by the sentinel, and the server where it runs. Additionally, it also requests the login credentials of the EHR. This ensures that data can only be extracted by the EHR system user. Providing correct login credentials activates data extraction process (see Figure 4.5), whilst inputting wrong login credentials abort the process and return an error message. Entering login credentials every time the physician wants to import data from the EHR was preferred over storing the login credentials in the application’s database, enhancing the security of the process.
After data extraction, MANAGE button allows users to view a list of all the cases obtained from the EHR system (see Figure 4.6). At this point, users have two options: displaying and completing data regarding a case (EDIT) or deleting the case (DELETE). Displaying data regarding a case presents a graphical form, automatically filled with data from the EHR regarding that case (see Figure 4.6). Users can edit and complete the form if any data are missing and then terminate the form by pressing the CONCLUDE button, which will save the alterations to the database. At any moment the user can save the form, through the SAVE button, and conclude it later. When a form is concluded, its status changes to 1 and it is ready to be sent. Furthermore, deleted cases are still stored in the database, with the status 9.

Sending forms to the Sentinela+ server is achieved through the SUBMIT button. Users also must enter their login credential to send data to the server. These login credentials comprise their sentinel identification number and a Sentinela+ password, attributed to each sentinel, ensuring only actual
sentinels can submit data. Only forms with status equal to 1 (concluded forms) are submitted. Once the notification form is submitted, its status is changed to 7.

Figure 4.7: Interface to request data sending to Sentinela+ server.

The whole process is summarized in Figure 4.8, where it is possible to observe the status forms go through.

Figure 4.8: Semi-automated reporting process in Sentinela+ client application.

4.2.4 Other Features

The main purpose of Sentinela+ client is to allow the semi-automated notification of cases, using data from the EHR. However, it comprises other features, described in the following sub-subsections.

4.2.4.A Manual Reporting

To fill a new form from scratch, without using data from the EHR system, users must utilize CREATE button. Users choose which clinical event they want to report and an empty form regarding that event is presented. The form is assigned a key and its status set to 0. Users can store the form data in the application’s database, through SAVE or CONCLUDE buttons. When a form is saved or concluded, a new row is created in the NOTIFICATIONS table in the local database. Data inputs performed through GUI elements are captured and stored in this row, as depicted in Section 4.2.2.
SAVE button allows users to save partially filled forms – incomplete forms – which remain with status equal to 0. These forms can be finished later or deleted using the EDIT feature. CONCLUDE button is used to save entirely filled forms – complete forms – changing their status to 1. These forms are ready to be submitted and can no longer be edited. When a form is concluded (status equals 1) it can be sent using SUBMIT feature. The whole process is summarized in Figure 4.9:

![Figure 4.9: Manual reporting process in Sentinela+ client application.](image)

### 4.2.4.B No Cases or Absence Notification

As mentioned in Subsection 3.3.7, sentinels must report periods when no cases are notified, either because there were none or the sentinel was absent from medical practice. Since these data cannot be extracted from the EHR systems, the process has to be performed manually. The application has a feature to facilitate this process, providing a GUI that allows users to choose the period of time without submission of notifications and the reason for the lack of notifications (Figure 4.10). These data are stored in a new row of the table NOTIFICATIONS on the local database, in the fields absence_week and absence_motive. The other fields in the row are all left empty, except for the notification_key and the notification_date. Even though the user introduces the date upper and lower limit for the period without notifications, these data are actually stored as the week numbers corresponding to that period of time, using the ISO week date standard conversion. A period of time can correspond to multiple weeks, thus a data entry may generate multiple data rows. For example, assume the user introduces the dates January 1 to January 7 and the motive for absence “vacation”. These dates encompass two different ISO weeks, week 1 and week 2. Thus, two data rows are created, one with absence_week set to 1 and other with this value set to 2, and both rows with the same absence_motive value. The reason for this is that each week without notification forms must be notified separately for the weekly incidence rate calculation.
4.2.4.C Extract and Send List of Patients

As mentioned in Chapter 3, sentinels must submit their list of patients in the beginning of each civil year, to define the population under observation. Sentinela+ client also makes possible to extract each sentinel list of patients from the EHR system and submit it to Sentinela+ server.

This process is identical to data extraction process regarding cases of notifiable events, described in subsection 4.2.3. Users provide login credentials and data processes are requested by the application. However, the information is directly submitted to Sentinela+ server and data are not stored in the application's database, since this process is only performed once a year.

The list of patients can already be exported from the EHR, and sent by paper or mail to INSA, where it is manually entered into a database system. This feature demonstrates how the task could be integrated within the application used to perform notifications.

4.2.4.D View Submitted Notification Forms

Users can also view notification forms that have already been submitted, which allows them to search all submitted forms, unfiltered or filtered by patient identification, event identification or notification date. This feature can be useful to keep track of the occurrences reported. Moreover, physicians can also search submitted forms by notification key. This latter should be used to double-check submitted data, if requested by MS system administrators.

4.2.4.E Alert Messages

The Sentinela+ desktop application comprises a system of alert messages, intended to provide reminders to sentinel users. When users start the application, three warning messages can appear.

- “You have notification forms to submit.” – Triggered if the sentinel did not submit concluded notification forms.
• “You have notification forms to conclude.” – Triggered if the physician extracted data and did not validate and conclude the respective form or if he saved an incomplete notification form that needs to be concluded.
• “You have not submitted any notification form for the period of a week.” – Triggered if the physician did not submit any data for the period of a week, redirecting him to the reporting absence of notifications feature (see Subsection 4.2.4.B).

These warning messages can be observed in Figure 4.11.

Figure 4.11: Warning messages in Sentinela+ application, displayed when the user runs the application if appropriate.

Warning messages can improve the timeliness of notifications, by giving physicians reminders of their notification activities.

4.2.4.F Error Messages

Furthermore, Sentinel+ application includes a series of error messages, aiming to make sure users fill notification forms correctly and completely. For instance, in diabetes cases, if a user associates gestational diabetes with a male patient, the application displays an error message saying that this is not possible, and does not let the user conclude the form (see Figure 4.12). Error messages can improve the quality and completeness of data collected.
4.3 Sentinela+ Server

4.3.1 Database

Completed notification forms in the client desktop application are submitted to Sentinela+ server, where data are stored in a database. The database holds data collected by sentinel physicians, which can be returned to MS system administrators.

This database comprises two tables **NOTIFICATIONS** and **ACCESS**. **NOTIFICATIONS** was created in the likeness of the client’s local database table with the same name, since it contains the same data. However, some minor changes were introduced. Two fields were added in the remote database structure: `id_sentinel` and `export_status`. The first one determines which sentinel physician submitted the form, which is obtained from the credentials provided when submitting data. The second one, establishes if the form was already exported or not by MS system administrators, through the web application described in 4.3.3. Moreover, the internal key (`internal_key`) and status of notification forms (`status`) in the client are not sent to the server, since these field are used to manage forms within the client application only. Also, patient identification (`id_patient`) is not sent to Sentinela+ server, since all data must be de-identified. Thus, the patient identification is only visible to sentinels within the application, allowing them to know to which patient each form refers to. When the form is submitted to the server, this data element is removed, guaranteeing that the server’s database comprises exclusively de-identified data.

**ACCESS** table comprises two fields: `id_sentinel` and `password`. This database was populated with the sentinel codes, for all the active sentinel physicians in the network, and a randomly generated password attributed to each sentinel. This table allows the authentication process when sending data to the server.
4.3.2 Data Upload Service

In order to insert data sent from the client in the database, a web service was developed, called SentinelaSubmit. This web service was developed on Visual Studio Express 2013 for Web, using code behind architecture in C# language. The web service was deployed in the IIS application server, on Sentinela+ server, and inserts data in the Sentinela+ database within the database server. The SentinelaUpload service can be invoked through the client, returning a response (see Figure 4.13).

![Diagram of client-server communication](image)

**Figure 4.13:** Client-server communication, between Sentinela+ client and Sentinela+ server.

The service comprises two web methods, Authenticate and InsertForm.

The Authenticate web method in SentinelaUpload determines if the login credentials provided through the client are valid, to ensure only sentinel physicians can submit data to the server. The web method is invoked by an HTTP POST request to with two parameters: `@username` and `@password`. This parameters are the credentials provided by the sentinel physician when submitting data (see Figure 4.7). The web method queries the database on the server and selects the number of rows in table ACCESS where the fields `id_sentinela` and `password` match the parameters received. The response is returned to the client and can be either 1 or 0. If the response is 1, it means there is one record in ACCESS for which the credentials provided match. Thus the login is authenticated and the client can now call the web method InsertForm and submit data. If the response is 0, the opposite occurs, and the login is not authenticated; the client returns an error message and no data are submitted.

The InsertForm web method allows to insert a single notification form data in the server’s database. It is invoked by an HTTP POST request from the client, which parameters contain all the fields in a single row of the local database table NOTIFICATIONS, except the fields `id_patient` and `status`, for the reasons described in Subsection 4.3.1. Additionally the sentinel identification number used in as a credential to send data is also sent as a parameter. When the web method is called, it inserts all the parameters received in the server’s database. The service returns an empty response, associated to a HTTP response code. This code is read by the client and gives user feedback according to the code received. If the returned code is 200, the standard response for successful HTTP requests, it means that the HTTP request was successfully executed. If so, the client changes the status of the notification form submitted (see subsection 4.2.3). While there are notification forms to submit, this whole process is repeated.
4.3.3 Datasets Download Web Client

Once notification forms de-identified data are stored in Sentinela+ server, MS system administrators need to be able to export these data to INSA server.

Sentinela+ comprises a web application, to return datasets from the Sentinela+ server in CSV format. The web application was hosted on the application server in Sentinela+ server and it communicates with the database server, within Sentinela+ server, selecting datasets to export and returning them to the user. To retrieve datasets, the web application queries the database and select all data rows that were not exported yet, which correspond to rows where the field export_status equals 0. Datasets are returned by clinical event as a CSV attachment.

The target users of the application are MS system administrators at INSA. The application can be accessed through a web browser. Users can select the clinical event to export through a drop-down list. The generated CSV is made available to download as an attachment. The user interface is illustrated in Figure 4.14.

Since no real patient's data was used in this project, access to the web application on the Web is open. If real data were to be used, it would be necessary to develop an authentication process, ensuring only MS system administrators could access the application.

CSVs are imported into a dBase DBMS, running of INSA servers. Importing process has to be performed manually, through dBase command line. Data from each CSV is structured in a way that allows direct importation into dBase software.

4.4 MedicineOne Connection

As mentioned in Section 2.4, two major EHR systems are used in NHS: SAM and MedicineOne. I requested test models of these EHR systems to the responsible entities, to develop an extraction process that could return data from these systems. Only MedicineOne was available for cooperation,
which determined that integration would be conducted exclusively with this EHR system, leaving SAM out of the equation by now.

Sentinela+ was connected to the MedicineOne EHR system, extracting data for cases of MS-notifiable health conditions. The Sentinela+ client communicates with MedicineOne server over the Web, through HTTP protocols, using a module called MedicineOne Connector. To bypass security firewalls, I created a web service and hosted it on MedicineOne server (MedicineOne-Sentinela web service). Through the Connector module, the client can invoke the web service, which extracts the necessary data from MedicineOne database, returning them to the client. The conceptualization of the Sentinela+ system with a connection to MedicineOne is illustrated in Figure 4.15.

![Figure 4.15: Conceptualization of Sentinela+ system, with a connection to MedicineOne.](image)

The MedicineOne-Sentinela web services were developed in ASP.NET framework, with code behind architecture in C# language with no user interface, using Microsoft Visual Studio Express 2013 for Web. MedicineOne server comprises a database server with Microsoft SQL Server 2008 and an application server with operative system Windows Server 2012 R2 and an IIS 8.0 web server.

### 4.4.1 MedicineOne Data Model

In this project I focused on extracting administrative data, diagnosis data and medication data from MedicineOne EHR system, which covers most of the data needs for MS notification forms. These data could be accessed from a small number of fields in MedicineOne database, scattered through seven tables. These tables and fields include:

- **CLI_UTENTES** – Administrative data regarding all patients.
  - **Pk.utente**: identifies the patient in the table; primary key.
  - **Np**: patient’s process number.
  - **Data.n**: patient’s date of birth.
  - **Sexo**: patient’s gender
  - **Pk.medico**: identifies the responsible physician for the patient; primary key in ORG.PESSOAS.
• **CLI_PATOLOGIAS_UTENTES** – Contains data on disease cases manifested by patients.
  – **Pk.patologia**: identifies the disease case in the table; primary key.
  – **Codigo.patologia**: ICPC2 diagnosis code for the disease case.
  – **Data.registo**: date when the disease case was registered.
  – **Pk.utente**: identifies the patient who manifested the disease; primary key in CLI_UTENTES.

• **CLI_PRESCRICOES UTENTES** – Contains data on prescribed medication cases for each patient.
  – **Pk.prescricao**: identifies the prescribed medication cases in the table; primary key.
  – **Pk.medicamento**: identifies the medication prescribed; primary key in table MEDICAMENTO.
  – **Data.registo**: date when the medication was prescribed.
  – **Pk.utente**: identifies the patient to whom medication was prescribed; primary key in table CLI_UTENTES.
  – **Pk.utilizador**: identifier for the MedicineOne user that prescribed the medication; primary key in table ORG_UTILIZADORES.

• **ORG_UTILIZADORES** – Contains data regarding MedicineOne users preferences and definitions.
  – **Pk.utilizador**: identifies each user in the table; primary key.
  – **Username**: username of the user.
  – **Password**: encrypted password of the user.
  – **Pk.pessoa**: primary key in table ORG_PESSOAS.

• **ORG_PESSOAS** – Contains administrative data regarding MedicineOne users.
  – **Pk.pessoa**: identifies each user; primary key.
  – **Nome**: name of the MedicineOne user.

• **MEDICAMENTO** – Contains the list of medications available for electronic prescription.
  – **Pk.medicamento**: identifies the medication; primary key.

• **ATC_LINK** – Contains the ATC codes for medications in table MEDICAMENTO.
  – **Pk.ATC**: ATC code for the medication; primary key.
  – **Pk.medicamento**: primary key in table MEDICAMENTO.

All the tables comprise foreign keys (FK) that are the primary keys (PK) of other tables, which allows to establish the relationship between them. For instance, table **CLI_PATOLOGIAS_ UTENTES** field **Pk.utente** is the PK on table **CLI_UTENTES**, thus it is possible to relate these tables using this field. Relations between the tables used to extract data can be observed in Figure 4.16.
4.4.2 Detection of Cases and Data Mapping

The first step to extract data was to determine how cases of MS-notifiable events could be detected in the MedicineOne EHR and which data regarding these cases could be extracted. The approach used to detect cases was based on diagnosis codes for notifiable diseases (Table 4.2) and on the ATC codes for anticoagulants prescription cases (Table 4.3). In MedicineOne database, ICPC2-coded differential diagnosis for a given consultation are stored in the field *Codigo patologia*, in the table *CLI_PATOLOGIAS_UTENTES*, which allows to detect MS-notifiable diseases. Additionally, all prescribed medications during a consultation are stored in the table *CLI_PRESCRICOES_UTENTES* and their ATC code can be obtained by relating it with the table *ATC_LINK*, which can be used to detect the anticoagulant prescription cases.

<table>
<thead>
<tr>
<th>Notifiable Diseases</th>
<th>ICPC2 Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILI</td>
<td>R80</td>
</tr>
<tr>
<td>Diabetes type 1</td>
<td>T89</td>
</tr>
<tr>
<td>Diabetes type 2</td>
<td>T90</td>
</tr>
<tr>
<td>Gestational diabetes</td>
<td>W85</td>
</tr>
<tr>
<td>Hypertension</td>
<td>K85, K86, K87</td>
</tr>
<tr>
<td>AMI</td>
<td>K75</td>
</tr>
<tr>
<td>CVA</td>
<td>K90</td>
</tr>
</tbody>
</table>

Table 4.2: ICPC2 codes for the MS-notifiable health events.

<table>
<thead>
<tr>
<th>Notifiable Anticoagulant</th>
<th>ATC Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acenocumarol</td>
<td>B01AA07</td>
</tr>
<tr>
<td>Apixaban</td>
<td>B01AF02</td>
</tr>
<tr>
<td>Dabigatran</td>
<td>B01AE07</td>
</tr>
<tr>
<td>Rivaroxaban</td>
<td>B01AF01</td>
</tr>
<tr>
<td>Warfarin</td>
<td>B01AA03</td>
</tr>
</tbody>
</table>

Table 4.3: ATC codes for the MS-notifiable anticoagulants.

Figure 4.16: Tables and fields used to extract data and the relationships between the tables.
After determining how cases could be detected, I determined the data elements in MedicineOne database that could be used to fill Sentinel+ forms, through data mapping. Data mapping is the process of creating links between data elements from distinct data models, and it is a fundamental component of data integration [37]. Through data mapping it is possible to specify how data structured under MedicineOne data model, the source data system, can be converted into data structured under Sentinel+ data model, the target data system. Data mapping is presented on Table 4.4.

<table>
<thead>
<tr>
<th>Sentinel+ NOTIFICATIONS</th>
<th>MedicineOne Table</th>
<th>MedicineOne Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_patient</td>
<td>CLI_UTENTES</td>
<td>Np</td>
</tr>
<tr>
<td>dob_patient</td>
<td></td>
<td>Data_n</td>
</tr>
<tr>
<td>gender_patient</td>
<td></td>
<td>Sexo</td>
</tr>
<tr>
<td>event_id</td>
<td>CLI_PATOLOGIAS_UTENTES</td>
<td>Codigo_patologia</td>
</tr>
<tr>
<td>event_date</td>
<td>CLI_PATOLOGIAS_UTENTES</td>
<td>Data_registo</td>
</tr>
<tr>
<td></td>
<td>CLI_PRESCRICOES_UTENTES</td>
<td>Data_prescriacao</td>
</tr>
<tr>
<td>oacp_treatment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbt_treatment1</td>
<td>ATC_LINK</td>
<td>Pk_ATC</td>
</tr>
<tr>
<td>dbt_treatment2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbt_treatment3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hyp_treatment1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hyp_treatment2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hyp_treatment3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oacp_initiative</td>
<td>ORG_UTILIZADORES</td>
<td>Username</td>
</tr>
<tr>
<td>dbt_initiative</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hyp_initiative</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oacp_initiativewho</td>
<td>ORG_PESSOA</td>
<td>Nome</td>
</tr>
<tr>
<td>dbt_initiativewho</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hyp_initiativewho</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbt_type</td>
<td>CLI_PATOLOGIAS_UTENTES</td>
<td>Codigo_patologia</td>
</tr>
</tbody>
</table>

Table 4.4: Mapping between data elements in Sentinel+ NOTIFICATIONS table and data elements in the MedicineOne EHR. This mapping allows to determine which fields in MedicineOne can be used to fill in Sentinel+ notification forms.

Since all the cases are detected in the MedicineOne tables CLI_PATOLOGIAS_UTENTES and CLI_PRESCRICOES_UTENTES, the other tables are accessed using their relation with those two tables (see Figure 4.16). The event_id in Sentinel+ is obtained from the field Codigo_patologia for notifiable disease cases and set automatically for anticoagulants prescription cases. Furthermore, the event_date in Sentinel+ is obtained from the field Data_registo for notifiable disease cases and from the field Data_prescriacao for anticoagulant prescription cases.

Moreover, in MedicineOne medications are not directly associated with the diseases they were prescribed for. It is possible to observe that notifiable disease cases can be detected on table CLI_PATOLOGIAS_UTENTES, but this table is not directly related to the table CLI_PRESCRICOES_UTENTES (see Figure 4.16), where medication prescription cases can be found. However, Sentinel+ forms require the medications prescribed for diabetes and hypertension disease cases. To overcome this problem, disease cases were associated with the medication prescribed through the
4.4.3 Extraction Process

Once I established how cases of notifiable events could be detected and how could data regarding these cases be accessed in the MedicineOne database, I developed a process to actually detects and extracts data. This was achieved through web services software.

A web service is a software system designed to support interoperable machine-to-machine interaction over a network [38]. It is a solution frequently used in system's integration, that makes information exchange across distinct systems possible, usually through Extensive Markup Language (XML) standardized messages. Thus, two systems can have different operating systems and programming languages and communicate through web services. In this particular case, a web service allowed the communication between Sentinel+ and MedicineOne.

I developed an ASP.NET web service using Microsoft Visual Studio Express 2013 for Web, using code behind architecture in C# language with no user interface. This web service was called MedicineOneSentinela and it was hosted on MedicineOne’s server, specifically on its application server. The web service comprises web methods, with the purpose of extracting data from MedicineOne’s database, within the database server, and return them to the client. Each method is called by an HTTP POST request from the client, with parameters enclosed within its body. The methods allow the connection to MedicineOne’s database, using Windows authentication, and execute a query filtered by the request’s parameters, producing a XML response with the extracted data. Figure 4.17 illustrates this process.

The web service comprises three web methods: Authenticate, ExtractDisease and ExtractAnticoagulant.

Authenticate web method determines if the login credentials provided by the user in the client are valid for a MedicineOne user. This web method is invoked by an HTTP request with two parameters: @username and @password, which are the login credentials provided when requesting an extraction process. The web method queries MedicineOne’s database, selecting the number of rows in table ORG_UTILIZADORES where the fields Username and Password match the parameters received.
This number equals 1 if the login credentials provided match a MedicineOne user, or 0 otherwise. A response is returned to the client as an XML, with the value obtained from the query. In the client, the response is read and if it equals 1, the other methods are called, which allows data extraction to take place; if it equals 0, an error message is returned to users and the extraction process is aborted, without any data being actually extracted (see Figure 4.18).

![Data extraction process diagram](image)

**Figure 4.18:** Data extraction process.

Additionally, *ExtractDisease* web method extracts data regarding reportable disease cases (ILI, diabetes, hypertension, AMI, CVA) and *ExtractAnticoagulant* data regarding anticoagulants prescription cases. These web methods are called by HTTP request with two parameters: ![](lastdate) date when the last exportation process was performed, stored in the client’s application database, and ![](username) MedicineOne login username provided when requesting data extraction. These parameters limit the extraction of data to registries where 1) event date is greater or equal to the parameter ![](lastexportation), 2) patients are strictly from the list of patients of the MedicineOne user ![](username).

*ExtractDisease* method queries MedicineOne’s database, selecting the patient’s identification (Np), date of birth (Data_n) and gender (Sexo), and the reportable disease code (Codigo_patologia) and diagnosis date (Data_registo), where the disease code corresponds to one of the ICPC2 codes of the reportable diseases. Additionally, it is also necessary to collect data regarding medications prescribed for diabetes and hypertension cases. These data include medication prescribed code (Pk_ATC) and the medication prescriber (Username, Nome). However, in MedicineOne’s data model, medications are not directly associated with the diseases they were prescribed for. To overcome this problem a second query is performed, which extracts any medication prescribed to patients who were diagnosed with diabetes and hypertension in the diagnosis date. Evidently, this query can return medications that were not prescribed for diabetes nor hypertension, for example if a physician prescribes medication for another health condition during a consultation where a patient was diagnosed with diabetes. These undesirable medications are handled in the client, as explained in section 4.4.4. Notice that this second query is limited by values obtained from the first query, including the patient’s identification and the diagnosis date.

On its turn, *ExtractAnticoagulant* collects data about anticoagulants prescription cases. This web method also extracts the patient’s identification (Np), date of birth (Data_n) and gender (Sexo), however it does not select disease codes and diagnosis dates. Instead, it selects the medication ATC code.
(Pk_ATC) and prescription date (Data_prescriacao), for medications where the ATC code matches the reportable anticoagulants. Additionally, this query also selects data about the physician who prescribed the medication, including its username (Username) and name (Nome).

The results of queries from these web methods are returned to the client in XML format. An XML document is a string of characters, comprising markups (or tags) that describe the data elements extracted. To build the XML, two classes were created within the web service, defining the XML tags and structure for ExtractDisease and ExtractAnticoagulant responses.

The correspondence between the data element extracted from MedicineOne and the markup in the XML response is presented in Table 4.5 and Table 4.6.

<table>
<thead>
<tr>
<th>MedicineOne Table</th>
<th>Column</th>
<th>XML Markup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLI_PATOLOGIAS_UTENTES</td>
<td>Codigo_patologia</td>
<td>&lt;disease_code&gt;</td>
</tr>
<tr>
<td>CLI_PATOLOGIAS_UTENTES</td>
<td>Data_registro</td>
<td>&lt;disease_date&gt;</td>
</tr>
<tr>
<td>CLI_UTENTES</td>
<td>Np</td>
<td>&lt;patient_id&gt;</td>
</tr>
<tr>
<td>CLI_UTENTES</td>
<td>Data_n</td>
<td>&lt;birthday&gt;</td>
</tr>
<tr>
<td>CLI_UTENTES</td>
<td>Sexo</td>
<td>&lt;gender&gt;</td>
</tr>
<tr>
<td>ATC_LINK</td>
<td>Pk_ATC</td>
<td>&lt;medication_code&gt;</td>
</tr>
<tr>
<td>ORG_UTILIZADORES</td>
<td>Username</td>
<td>&lt;prescriber_id&gt;</td>
</tr>
<tr>
<td>ORG_PESSOAS</td>
<td>Nome</td>
<td>&lt;prescriber_name&gt;</td>
</tr>
</tbody>
</table>

Table 4.5: Mapping between MedicineOne database fields and XML response markups for ExtractDisease web method.

<table>
<thead>
<tr>
<th>MedicineOne Table</th>
<th>Column</th>
<th>XML Markup</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATC_LINK</td>
<td>Pk_ATC</td>
<td>&lt;medication_code&gt;</td>
</tr>
<tr>
<td>CLI_PRESCRICOES_UTENTES</td>
<td>Data_prescriacao</td>
<td>&lt;prescription_date&gt;</td>
</tr>
<tr>
<td>ORG_UTILIZADORES</td>
<td>Username</td>
<td>&lt;prescriber_id&gt;</td>
</tr>
<tr>
<td>ORG_PESSOAS</td>
<td>Nome</td>
<td>&lt;prescriber_name&gt;</td>
</tr>
<tr>
<td>CLI_UTENTES</td>
<td>Np</td>
<td>&lt;patient_id&gt;</td>
</tr>
<tr>
<td>CLI_UTENTES</td>
<td>Data_n</td>
<td>&lt;birthday&gt;</td>
</tr>
<tr>
<td>CLI_UTENTES</td>
<td>Sexo</td>
<td>&lt;gender&gt;</td>
</tr>
</tbody>
</table>

Table 4.6: Mapping between MedicineOne database fields and XML response markups for ExtractAnticoagulant web method.

The structure of the XML response generated by each web method is presented below. Notice that data regarding medication in ExtractDisease XML response, within the <ArrayOfMedication> markups, is only extracted for diabetes and hypertension cases.

**ExtractDisease XML response structure**

```xml
<ArrayOfDisease xmlns="http://tinyurl.com/qf485x4">
  <disease>
    <disease_code>string</disease_code>
  </disease>
</ArrayOfDisease>
```
Furthermore, a very similar web service was developed, to handle the extraction of the list of patients of the sentinel physician, described in subsection 4.2.4. It comprises only one web method, designed to select data on the patient’s year of birth and gender only. Since this web service is identical to the ones described and this was a minor feature, I will not describe it any further.

4.4.4 MedicineOne Connector

The extraction processes in the server must be invoked from the client. Additionally, the client also needs to read the XML response, convert its data and store these data in the local database. To handle these processes, I developed a connector module in the client. The connector module determines:

- The EHR system data are extracted from. In this project only a MedicineOne connector was
developed.

- The HTTP POST requests and the server to which they are sent.
- The username and password used to access EHR data, entered by users through the connector interface.
- How the response is processed and stored in the database.

![Connector interface in the client application](image)

**Figure 4.19:** Connector interface in the client application. This module allows users to request data extraction, sending HTTP requests to the chosen server, and processing the response received.

When users click the IMPORT button in the client, the connector module is called. Its interface allows users to enter the clinical system’s server, username and password. Then, the connector sends HTTP POST requests to the selected server, in order to accomplish data extraction. The first request parameters are the username and password entered by users, which allow to authenticate these login credentials, as described in 4.4.3. If the parameters are authenticated successfully, the connector proceeds to send two more HTTP POST requests, one for disease cases extraction and other anticoagulant cases extraction. The parameters sent in these requests are the username, previously entered by users, and the last extraction process date, stored in the client local database, in the table ACTIONS.

The server responds to the requests, sending data back from the clinical system. Data mapping between MedicineOne data elements in XML responses and Sentinela+ data elements (see Tables 4.4, 4.5 and 4.6) determines which data elements in Sentinela+ can be obtained from the XML. The conversion of XML data elements to Sentinela+ data elements is further depicted in the following paragraphs.

Extracted data are processed by the connector, by reading the response XML file and converting its data to a format that can be stored in the client local database. Data elements within the XML are read using the corresponding markup name, using Java function `getElementsByTagName()`. The data element within the markup is obtained and converted to the desired format.

*ExtractDisease* XML response comprises a list of reportable disease cases extracted, and each one is defined by the markup `<disease>`, which is the root for the five data elements extracted regarding the disease. For each disease, a new row is created in the table NOTIFICATIONS in
Sentinela+ client local database. The element with <disease_code> markup comprises the ICPC2 code for a reportable disease and it is used to determine the event_id, using the following correspondence: R80 – 1; T89, T90, W85, T99 – 2; K85, K86, K87 – 3; K75 – 4; K90 – 5. The data elements with <disease_date> and <birthday> markups are used to fill the fields event_date and dob_patient, after parsed from string to date format. Additionally, elements with markup <patient_id> and <gender> are used to fill the fields id_patient and gender_patient, after parsed from string to integer. Additionally, each medication extracted for diabetes and hypertension cases is represented by the markup <medication>, which are comprised within the markup <ArrayOfMedication>. This list of medications comprises all medications prescribed to a patient diagnosed with diabetes and hypertension in the diagnosis date. The medication code in the XML response, within the markup <medication_code>, is read and if it corresponds to a listed medication for diabetes and hypertension, it is stored in the fields dbt_treatment(1-3) or hyp_treatment. If it does not, it is ignored. Additionally, the data element with markup <prescriber_id> comprises the MedicineOne username of the medication prescriber. This data element is used to fill the field initiative in Sentinela+ application database, by comparison with the username entered when requesting the extraction process. If both usernames match, then initiative value is set to 1 (the sentinel physician prescribed the medication); otherwise, it is set to 2 (other physician prescribed the medication) and the name of the prescriber is stored in the field initiative_who, obtained from the data element with markup <prescriber_name>.

ExtractAnticoagulant XML response structure comprises a list of anticoagulant prescription events, each one defined by the markup <anticoagulant>. For each anticoagulant a new row is created in table NOTIFICATIONS. When this response is received, the application sets the event_id in the rows to 6 by default. The data elements with markups <prescription_date> and <birthday> are used to fill the fields event_date and dob_patient, after parsed from string to date format. Similarly, data elements with markup <patient_id> and <gender>, are used to fill the fields id_patient and gender_patient, after parsed from string to integer. Data elements with markup <medication_code> are used to fill the field oacp_treatment. Data elements with markup <prescriber_id> and <prescriber_name> are used to fill the fields initiative and initiative_who, through the same process described for ExtractDisease XML response for the markups with the same name.

4.4.5 Summary

To summarize, Sentinela+ introduces a new reporting method in the MS sentinel network, which resorts to data from clinical systems for automated filling of the notification forms at use. Sentinela+ encompasses a client, which is a tool to fill the forms, and a server, to where forms data are sent. Additionally, it includes a web client, to download datasets from the server, by MS system administrators. Datasets can be imported into INSA server, for further processing and analysis, for public health purposes.

At the moment, Sentinela+ provides a connection to the MedicineOne EHR system. Furthermore, the system was not deployed in primary care centres yet, which is the needed step to start using Sentinela+ to notify real patient data.
Contents

5.1 User Satisfaction Evaluation .................................................. 58
5.2 Performance Evaluation ......................................................... 61
5.3 Summary ............................................................................... 64

Sentinela+ Evaluation
5.1 User Satisfaction Evaluation

To ascertain Sentinela+ target users satisfaction – the sentinel physicians in MS network – a parametrized questionnaire was applied. This questionnaire was an adapted version of the Questionnaire for User Interaction Satisfaction (QUIS). QUIS was designed to assess users’ subjective satisfaction regarding aspects of the computer interface and usability. This questionnaire was developed by a team of researchers in the Human-Computer Interaction Laboratory at the University of Maryland in 1987 and it is currently at version 7.0. QUIS is a measure that is highly reliable across many types of interfaces [39]. QUIS comprises two sections: a measure of overall satisfaction with the system and measures of satisfaction in four specific interface aspects of the system.

The questionnaire was handed to eight sentinel physicians, of which four answered the questionnaire. Thus, the sample size was four, which represent 6% of the active sentinels and 3.8% of the entire number of sentinels.

The measure of overall satisfaction with the system consists of 6 ratings with a 9-point scale. The lowest and highest values on the scale are associated with a word to describe user assessment. These words are: 1) terrible–wonderful 2) difficult–easy 3) frustrating–satisfying 4) inadequate power–adequate power 5) dull–stimulating 6) rigid–flexible. Responses given by the users can be seen in Figure 5.1.

![Overall Reaction to the Software](image)

**Figure 5.1:** Questionnaire responses regarding the overall satisfaction with Sentinela+ system.

It is possible to observe that 3 of the 4 users queried demonstrated an high overall satisfaction with
Sentinela+ system. The other user overall evaluation of the system was positive, but encompasses more neutral ratings. However, all users agreed that system was satisfying and with an adequate power.

Evaluation of satisfaction with the interface of the system includes measures for four elements: 1) screen components, 2) terminology and system feedback, 3) learning factors, 4) system capabilities. There is a section for each element, which measures factors regarding that element of the interface, also using a 9-point scale where values are associated with words to ascertain user’s assessment. The factors evaluated for each interface element are enumerated in Table 5.1.

<table>
<thead>
<tr>
<th>Interface element</th>
<th>Evaluated factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen</td>
<td>Readability of characters on the screen</td>
</tr>
<tr>
<td></td>
<td>Organization of information</td>
</tr>
<tr>
<td></td>
<td>Sequence of screens</td>
</tr>
<tr>
<td>Terminology and system feedback</td>
<td>Use of terminology related to the task</td>
</tr>
<tr>
<td></td>
<td>Prompts for input</td>
</tr>
<tr>
<td></td>
<td>Computer informs about progress</td>
</tr>
<tr>
<td></td>
<td>Error messages</td>
</tr>
<tr>
<td></td>
<td>Position of messages on screen</td>
</tr>
<tr>
<td>Learning</td>
<td>Learning to operate the system</td>
</tr>
<tr>
<td></td>
<td>Remembering names and use of commands</td>
</tr>
<tr>
<td></td>
<td>Performing tasks is straight-forward</td>
</tr>
<tr>
<td></td>
<td>Help messages on the screen</td>
</tr>
<tr>
<td></td>
<td>Supplemental reference materials</td>
</tr>
<tr>
<td>System capabilities</td>
<td>System reliability</td>
</tr>
<tr>
<td></td>
<td>System speed</td>
</tr>
</tbody>
</table>

Table 5.1: Evaluated factors in each interface element.

Users’ evaluations regarding elements of Sentinela+ interface, in the client desktop application, can be observed in Figure 5.2. In relation to the screen, 3 out of the 4 users agreed that reading characters in the screen is easy. While two of the users evaluated the organization of information in the screen as very clear, the two other users did not find it that clear. The users agreed that there is room to improve the sequence of screens, even though they evaluate it positively. Overall, the screen components were evaluated positively, but requires some improvements. Concerning system’s capabilities, it is a consensus amongst users that the system is fast and reliable. Regarding the learning process of the system, 3 out of the 4 users found the application easy to learn. Additionally, all users agree that performing tasks is straight-forward and supplemental reference materials are clear. However, two users reported that it was not easy to remember the name of the functions and use of commands. At last, in relation to terminology and system feedback, 3 out of 4 users evaluate it very highly overall, while one user evaluates it as neutral.
Figure 5.2: Questionnaire responses regarding the user satisfaction with specific interface aspects of Sentinela+ client.

To complement the evaluation, an open question was applied, for users to enumerate Sentinela+ most positive and negative aspects. Users referred simplicity, easiness to operate, speed, visual attractiveness and helpful error messages as the strong points. Negative aspects included characters small font size and the lack of task progress messages when the system goes through loading processes (e.g. when the system’s client requests an extraction process and waits for the response). Users also referred that they could not run the client in their office computers, due to the outdated software technology installed in these computers.

The QUIS does not include measurements of specific functions of systems, since these depend too much on the particular system. Thus, some additional measurements were created to evaluate how users perceive the value of the system to MS network. These evaluations were also performed through ratings with a 9-point scale, where values within the scale are associated with adjectives. Two factors were evaluated: how the system facilitates the notification process and how it diminishes the time spent on the notification process. The results obtained can be observed in Figure 5.3.
The users evaluated the system value very positively. It is unanimous that the system facilitates the notification process significantly. Moreover, 3 out of the 4 users also agree that the system diminishes the time spent in the notification process significantly.

Additionally, two yes/no questions where applied in the questionnaire, to further assess how the system could be valuable for MS. These questions aimed to determine if 1) the physicians would use Sentinela+ system instead of the current reporting tools (RIOS-MS web application and paper), 2) the physicians feel that Sentinela+ system could attract new physicians to join MS network. All the inquired physicians responded unanimous and affirmatively to both questions.

5.2 Performance Evaluation

To evaluate Sentinela+ performance I resorted to JMeter, an open-source tool to load test functional behavior and measure performance. JMeter can be used to simulate a heavy load on a server to test its strength [40]. I tested data extraction from MedicineOne server and data ingestion into Sentinela+ server. MedicineOne server has an Intel Xeon CPU E5520 2.27 GHz, 1.67GB RAM, running Windows Server 2012 R2 as the operating system. Sentinela+ server an Intel Xeon CPU E5-2670 v2 2.50 GHz, 1GB RAM, running Windows Server 2012 R2 as the operating system. For these tests I used a machine with an Intel Core i7-3537U 2.00 GHz processor, 8GB RAM and network with symmetric bandwidth.

I simulated sending notification forms data into Sentinela+ server, through HTTP POST requests, by a variable number of users. The tests aim to observe how the system responds to different stress levels. The performance indicator collected was the response time, i.e. the time the server takes to accept a notification form. I ran tests with 100, 120, 150, 200, 500 and 1000 users submitting data into Sentinela+ simultaneously. Simulations were looped 100 times, and the average response time computed. This high number of loops allows to get an average response time that minimizes the external effect in response times of elements within the network. The results are presented in Figure 5.3.
Figure 5.4: Average response time versus the number of users, for data sending processes.

For the current number of maximum possible users (107) are in the response times are in the order of 400ms. It is also possible to observe that response times increase linearly with the number of users, as expected. This allows to predict well how the system need to be scaled as the number of users (sentinels) increase. For instance, doubling the number of users would have little impact on the average response time of the system, thus the system would not require any change to support this increase.

Additionally, I present all the response times obtained for 100 users during the simulation period (Figure 5.5) and the distribution of those response times (Figure 5.6).

Figure 5.5: Response times over the duration of a data sending simulation, for 100 users.
Figure 5.6: Distribution of the response times obtained over the duration of a data sending simulation, for 100 users.

It is possible to notice in Figure 5.6 that response times fluctuate and reach high peak values, which represent abnormal response times due to effects of factors within the network used to perform the tests, as already mentioned. Normal response times lie between 393ms and 496ms, as can be observed in Figure 5.6. The average response time per request was 436ms.

I also simulated the extraction of data from MedicineOne EHR system by a single user, measuring the response time. The simulation was looped 100 times.

Figure 5.7: Average response times over the duration of a data extracting simulation, with 1 user.

Response times averaged of 57.75ms. This test was performed for the extraction of 10 cases from the EHR system. Further testing allowed to determine that this response time was not significantly affected by the number of cases extracted.

Additionally, I observed how data extraction affected MedicineOne server.
CPU usage raises from 1-2 % to values up to 8%, as can be seen in Figure 5.8. Although not negligible, these values allow the server to perform in good condition. Thus, Sentinela+ requests shall not affect the performance of the EHR system significantly.

Note that the results obtained would not be exactly equal if real users were using the system in a real case scenario, since in this test the requests were being fired from a single machine. Additionally, the results depend upon the machine and the network used. However, this simulation provided valid indicators to ascertain the system performance.

5.3 Summary

Sentinela+ was evaluated by users and through performance measures.

A sample of MS sentinels tested the system and evaluated through an adapted version of QUIS, a questionnaire for user satisfaction. Feedback received was positive. Users rated how the system was valuable for the notification process in two aspects: facilitation of the notification process (average score: 8.0 out of 9) and reduction of the time necessary to fill notification forms (average score: 7.3 out of 9). Additionally, users also rated four user interface aspects: screen (average score: 7.3 out of 9), terminology and system feedback (average score: 7.9 out of 9), learning (average score: 8.1 out of 9) and system capabilities (average score: 8.5 out of 9).

Performance tests consisted of simulations to test Sentinela+, which demonstrated the system performance was adequate. Response times were measured, and they were in the order of 50ms for 1 user extracting 10 cases data extraction and in the order of 400ms for 100 users (approximately the number of sentinels) sending data.
6.1 Conclusions

The main goals of this project were achieved. I was able to develop a fully functional system, Sentinela+, that extracts data from clinical systems, for automated filling of notification forms, to be used in a sentinel surveillance network, Médicos Sentinela. At the moment, Sentinela+ provides a connection to the MedicineOne EHR system. The project focused primarily on the development and implementation of all the components that allowed to have this functional system: the client application for automated form filling, the MedicineOne-Sentinela web service for data extraction from the MedicineOne database and the Sentinela+ server and its components, for further data processing.

Sentinela+ was able to extract most of the data regarding the cases of MS-notifiable diseases. The detection of cases was based on the ICPC2 diagnoses code for the notifiable diseases and on ATC codes for anticoagulants prescription. After detection, data regarding those cases where extracted, including administrative data, diagnosis data and medication data. Collecting medication data regarding the detected disease cases was the more challenging step. In MedicineOne, medications prescribed were not associated with the disease they were prescribed for. Thus, it was necessary to cross diagnosis dates and prescription dates, and then apply a filter to select the possible medications for the notified disease cases. Moreover, it was not possible to extract data regarding disease’s symptomatology, necessary for ILI cases, since there are no specific fields in the MedicineOne database to store disease’s symptoms. Since a great focus was given to the development of the system components, data extraction algorithms can be refined in future versions and more data can be obtained from the EHR for the automated reporting in MS.

Sentinela+ was evaluated for user satisfaction evaluation and performance evaluation. User satisfaction evaluation was performed through the QUIS, a parametrized questionnaire used to measure user overall satisfaction with a system and its interface. Additionally, the questionnaire was adapted to include an evaluation of the system value to MS network. Analysis of the responses allowed to conclude that:

1. 3 out of the 4 users queried manifested an overall high general satisfaction with the system and its interface; the other user rated the system with a positive, yet more neutral evaluation.
2. All users agreed Sentinela+ substantially facilitates the notification process.
3. 3 out of the 4 users agree that Sentinela+ can diminish the time spent in notification process significantly.
4. All users would use Sentinela+ as a reporting tool instead of the current available ones, if they could.

Performance evaluation proved that the system configuration used to develop Sentinela+ has an adequate performance to support MS notification process.

For all the aforementioned results, I can say that Sentinela+ achieved the successful integration of a sentinel network and an EHR system, and proved that this integration is indeed viable and beneficial. The system conceptualization and implementation can provide useful insights for the integration of
other sentinel and EHR systems. Additionally, Sentinela+ successfully integrated clinical and public health systems, which is a major challenge in the public health informatics domain.

6.2 Future Works

I propose a connection to SAM, the other major EHR system in the NHS, in Sentinela+. This connection would be extremely relevant, since most GPs in national primary care centres use this clinical system. To achieve this, it would be necessary to map Sentinela+ and SAM data elements, develop a SAM connector module for the client application to handle data extraction processes and implement an extraction process to collect data from SAM and return them to the client.

Furthermore, I propose the implementation of Sentinela+ system in a primary care centres using the MedicineOne EHR system, which implies installing the developed Web service for data extraction in the servers of those primary care centres. However, logistical issues represent an obstacle to this process, given that installing new software requires the approval from the respective regulatory authorities. Furthermore, Sentinela+ deals with extremely sensitive data, which raises concerns regarding data privacy and security. Even though data extraction can only be performed with the login credentials to access the EHR, it would be necessary to review privacy and security policies and procedures in a real case scenario, such as the use of Secure Sockets Layers (SSL) for communication between the client application and the EHR.

Once deployed in primary care centres, Sentinela+ must be subjected to a long-term evaluation. This evaluation should test the sensitivity of the system, by measuring the proportion of notifiable cases that are detected as such. Additionally, a study on the impact of the system in MS is also relevant. This study should answer questions such as: 1) Was the adherence to Sentinela+ as a notification tool high or low? 2) Did Sentinela+ increased the average number of weekly notifications? 3) Did the system attract new physicians to join MS?

I had the opportunity to verify that computers in some primary care centres use outdated and discontinued Java technology, which prevents Sentinela+ client application to run in these computers. Thus, in some primary care centres updates to the base software of desktop office computers would be necessary.

The Sentinela+ client was designed to run in the sentinels personal laptops, connected to the private networks of the primary care centres. However, with the increasing adoption of mobile technology, such as smartphones and tablets, a mobile version of the application could provide a more ergonomic alternative to this reporting method. Thus, the development of a mobile application as an extension for Sentinela+ system would be a great addition. Sentinela+ desktop application could serve as the prototype for the GUI of the mobile application. Moreover, the desktop application code could easily be recycled for an Android application, since applications written for this operative system frequently use Java-based code as well. By adhering to standard formats and protocols, the rest of the system adopted for Sentinela+ developed in this project would remain unchanged.
Bibliography


Sentinela+ Notification Forms
**Figure A.1:** ILI notification form in Sentinela+ client application.

**Figure A.2:** Diabetes notification form in Sentinela+ client application.
Figure A.3: Hypertension notification form in Sentinela+ client application.

Figure A.4: CVA notification form in Sentinela+ client application.
Figure A.5: AMI notification form in Sentinela+ client application.

Figure A.6: Anticoagulant prescription notification form in Sentinela+ client application.