

PHIS: The Philippine Health Information System

Critical Challenges and Solutions

by Ruben D. Canlas Jr., MSIT, MBA

rcanlas@alumni.cmu.edu

ABSTRACT

This survey research paper describes key technological challenges and solutions confronting the development of the Philippine Health Information System (PHIS).

PHIS is an online system that is intended to facilitate gathering, analysis and dissemination of vital health metrics. Through PHIS the Department of Health (DOH) increases access to health metrics data for the consumption of different users like planners, policy makers, community leaders and development agencies. PHIS therefore supports the important goal of equitable distribution of health services in the country.

Building PHIS means tackling major technological challenges on interoperability, governance and privacy. Technical interoperability is the ability to work with different and often incompatible databases maintained by various government agencies. Semantic interoperability is the ability that will allow PHIS to translate from one coding standard to another automatically.

This study recommends designing PHIS based on principles of enterprise architecture (EA) and service oriented architecture (SOA). It discusses salient principles of architecture, governance and privacy that will be useful starting points for designing, developing, and running PHIS. To supplement its case, this paper cites real-world projects that demonstrate the feasibility of such solutions.

Keywords

Health information system, service oriented architecture, ehealth, ontology

1. INTRODUCTION

The Philippines needs to create an integrated national health information system (HIS) to help improve the delivery of health services to citizens. Intended for planners, policy makers and decision makers, the Philippine Health Information System (PHIS) involves integrating data from various sources in different government agencies. The lead agency, Department of Health (DOH), needs assistance in understanding and solving technological challenges for building the HIS. This paper describes the critical technical challenges for such a project and proposes solutions and guidelines for improving the HIS design and implementation.

1.1 The Need for a National Health Information System

PHIS is an integrated electronic system for storing and sharing vital health indicators. PHIS will facilitate gathering, analyzing, and dissemination of health indicators to support better decisions,

enable timely response to public health emergencies and improve access to health.

PHIS will be the primary, authoritative source of national health metrics. It will also help improve the nation's participation in the global effort to improve health and mitigate global emergencies like pandemics.

1.2 Current Situation

PHIS data sources come from several government agencies: the National Statistical Office (NSO), National Statistical Coordination Board (NSCB), Food and Nutrition Research Institute (FNRI), National Nutrition Council (NNC), and Philippine Health Insurance Corporation (PhilHealth), among others. NSO provides basic census and population data like birth and mortality rates, while FNRI and NNC are sources of information on nutrition. PhilHealth is a source of patient records and NSCB is an aggregator of information to produce vital statistics like national economic and population metrics.

To facilitate inter-agency coordination, DOH created the Philippine Health Information Network (PHIN) in 2007, a coalition of agencies owning data to be used by PHIS (hereafter, "source agencies" or "partner agencies"). Through consultations, seminars and workshops, PHIN has built the needed linkages among the partner agencies. It has also assessed the current state of health information systems and formulated a general strategy for PHIS development (Philippine Health Information Network, 2007). The coalition also recently drafted a Data Dictionary containing definitions agreed by partner agencies of indicators, procedures, and terminologies relevant to PHIS. The Data Dictionary is available at <http://umis.doh.gov.ph/standards>.

1.3 Problem Statement

Although PHIN has made progress in defining PHIS from a conceptual level, it needs assistance in finding solutions to its major technical problems -- how to make disparate and incompatible information systems work together at less cost and effort, and how to resolve governance and privacy issues.

The problem can be stated thus: "What technological solutions will enable PHIS to work with existing IT systems and future services while minimizing the cost and effort involved?"

DOH recognizes that the technical solutions require an interoperable infrastructure. It has listed interoperability as one of its key result areas (Philippine Health Information Network, 2007).

2. SCOPE

2.1 Target Readers

The target readers for this paper are decision makers, in particular, government officers spearheading PHIS. The second target readers are software developers at the IT offices of the various agencies. These developers may have some familiarity with but little background on working with SOA and web services. The discussions in this paper need to appeal to these two reader types -- not too technical for the government officers, but with enough details for the developers to pursue.

2.2 Topic Focus

The focus is on various aspects of interoperability: of data, applications and processes. This paper proposes to build PHIS based on a service oriented architecture (SOA), to allow different participants to collaborate through an online system, lowering cost and effort. It also addresses critical issues about governance, security and implementation strategies.

2.3 Limitations of the Study

Time and scope. Creating a master plan for PHIS encompasses several domains of knowledge: public health, policy, software development, change management and even linguistics. Limiting the study to only a few of these domains would hamper thoroughness; covering as many domains as possible would be unrealistic. To achieve focus, the study concentrates only on critical issues on a macroscopic perspective and discusses other technological issues and solutions in broad strokes.

Availability of information. As the paper was being written, a series of typhoons and floods hit the Philippines, making the contacts from DOH busy with disaster response activities. This slowed down communication between the researcher and the contacts and also limited the amount of information available. Access to a good library was also difficult; the collection at CMU Adelaide is limited. The author worked around this by using the online collection of the CMU library, the local and state libraries of South Australia (which provided free access to EBSCO journals and articles) and a paid subscription to Safari Books Online (which provided access to developer references).

Proximity. Since the partner agencies are all based in the Philippines, a more detailed analysis of their existing setup and requirements is not possible. The requirements written in this paper are based on documents from DOH, HMN, the PHIN, and the participation of the author in meetings, workshops and correspondences with DOH, PHIN and other consultants.

2.4 PHIS Objectives

According to HMN, a national health information system's goal is to "increase availability, accessibility, quality and use of health information vital for decision-making at country and global levels." (Health Metrics Network, 2007, p. 1). The problem of access to basic health services and expertise is compounded by the fact that the Philippines is an archipelago of over 7,100 islands with doctors and nurses concentrated in urban areas (Philippine Health Information Network, 2007, pp. 12-13). An electronic system may provide a way for remote regions and communities to gain access to health experts that are usually concentrated in urban areas. HMN emphasizes that the desired outcome for such an electronic system is evidence-based policies and decisions and improved collaboration. The design of PHIS must therefore be aligned with these goals.

2.5 Design Goals

PHIS needs to be designed so it could provide accessible and reusable IT services for gathering, processing and disseminating health metrics.

To achieve this design goal, PHIS needs to facilitate technical and semantic interoperability. Technical interoperability is the ability to exchange information among disparate applications. Semantic interoperability allows computers to translate one coding standard to another without human intervention. Service oriented architecture and ontologies help solve these problems. These are discussed in the next sections.

3. MAJOR CHALLENGES

The critical challenges for PHIS are technical interoperability, flexibility, governance, privacy policies and semantic interoperability (Philippine Health Information Network, 2007, p. 18). The first four issues are addressed by software architecture and technology derived from existing industry practices. Semantic interoperability is a relatively young field involving the use of web services and ontologies. This section provides a background for understanding the challenges that need to be surmounted through technological solutions.

3.1 Technical interoperability

The Health Metrics Network reports, "Health information systems have evolved in a haphazard and fragmented way as a result of administrative, economic, legal or donor pressures. (Health Metrics Network, 2007, p. 6)." Non-standard data formats, applications and processes make it difficult for different entities to collaborate and share digital information without obstruction. The disparity is a result of technological constraints and the business strategy of vendors and clients in the early days of computing.

The goal of early software developers was to build applications in a computing environment constrained by limited memory and storage capacity. Developers focused on producing well-written procedures (or program logic) to run well in those environments. Software companies also strove to protect their businesses by imposing proprietary formats to block out competition.

As a further complication, client companies deployed applications in an ad hoc manner, usually through separate departmental initiatives. These computerization efforts typically started from easily automated business processes -- accounting, billing and payroll -- often using products from different vendors. These led to the following problems:

- Applications were tightly coupled with data storage. This prevented different applications or entities from using each other's data. Example: data stored in MS Access, is stored in .mdb format and cannot be used directly by MySQL (without performing a conversion). This was called the "information islands" problem in reference to the difficulty to access information stored inside the "islands" of software.
- IT services were tightly coupled with the applications created to deliver them. As a result, different departments and organizations ran several applications with duplicate data and features. Example: the Accounting Department used Excel to do statistical analysis of its data, while the Finance Department had to commission a custom-built software in Java to perform similar statistical procedures for its reports. As a result, departments in one organization often com-

peted with each other for limited resources. Moreover, this problem also existed on an inter-organizational level.

A quick inventory of the systems running at DOH illustrates the interoperability problem. The health department uses the following database products within its organization: MySQL, Sybase, PowerBuilder, MS Access, MS Excel, and FoxPro. DOH runs websites based on open source content management systems (CMS) Drupal and Joomla, but also has intranets in LotusNotes. In its websites, even though the underlying platforms are the same (Apache, MySQL and PHP), the sites use different database structures, workflows and user interfaces. DOH also has geographic information services (C. Tan, Head of the Knowledge Management Team at DOH, email correspondence, 6 November 2009).

The problem multiplies when DOH attempts to link with an external partner like the National Statistical Coordination Board (NSCB). NSCB web applications run on Microsoft Active Server Pages (ASP). How can PHIS harness these existing online systems without resorting to drastic measures like major code rewrites and even replacing total platforms? Moreover, as business needs change, data definitions often locked inside databases also need to be modified.

In recent years, several solutions relevant to PHIS have gained support -- XML, enterprise architecture (EA), service oriented architecture (SOA), ontologies and mesodata. These solutions are explained in detail in the next section.

3.2 Flexibility

Health metric reports need to be customizable for different types of users. Users from a national perspective will need less detail and more aggregation than those from subnational levels and global users will need even less detail and even more aggregation. Subnational users include local government officials, health workers, and hospitals and healthcare organizations (Health Metrics Network, 2007). National users include cabinet leaders, policymakers and planners. Global users may be regional development organizations (eg ADB) and international organizations like WHO.

3.3 Governance and Privacy Policies

PHIS governance issues may be divided into two categories: how the project will be governed by the different partners, owners of data sources and end users (hereafter, "project governance"); and the set of policies that governs the service oriented HIS (hereafter, "service policies").

Privacy is a major issue in the Philippines and must be addressed because it impacts data integrity and security. To avoid duplication of patient records, each citizen could be assigned a unique identification code (UID). In the Philippines, giving everyone a UID is a contentious issue. Having been under a dictatorship for more than twenty years, opposition political parties and activists have strongly opposed a national identification system that could have become the basis for UID (Napallacan, 2008; Burgonio, 2008).

3.4 Semantic Interoperability

Pencheon (2006, p. 78) captures the problem of semantic interoperability in the following excerpt from the Oxford Handbook of Public Health Practice:

Clinicians are used to categorizing people as either having or not having conditions, but in practice the boundaries of almost all diseases are unclear, and a full range of severity exists from the hardly perceptible to the catastrophic... The somewhat arbitrary nature of the definitions of the

boundaries of health states results in much of the variability in routinely collected data: without meticulous definition and attention, classification errors and biases abound.

Physicians may use a variety of ways to record a disease. Computers have no built-in way to know that a "migraine", for example, is the same as a "chronic severe headache" and this could lead to double-counting. This variance in representing diseases is greater across cultures and nations, thus presenting a serious obstacle to aggregating information at higher levels of abstraction.

Fox, Sahay, and Hauswirth (2009, p. 130) cite a "lack of interoperability within healthcare standards. (eg., HL7)" Different efforts to standardize data encoding have resulted to more incompatible data formats, creating obstacles for aggregating data at national and global levels (Health Information Network, 2007, p. . Countries also have different ways to encode public health data, like mortality rates, infant birth rates, and other public health measures. The Oxford Handbook of Public Health Practice emphasizes this problem: "There has yet to be an international classification for public health in the same way as we have the the International Classification of Diseases (ICD). (Weinberg and Pencheon, 2006, p. 197)"

Coding standards for diseases vary enough to confound automatic reading by computer software. In addition to ICD, there are SNOMED-CT (Systematized Nomenclature of Medicine-Clinical Terms), LOINC (Logical Observation Identifiers Names and Codes) and UMLS (Unified Medical Language System). Even ICD has to be adapted to different countries like Australia and United States. Some standards like SNOMED do not even cover all languages in such specializations as oral healthcare in the Swedish language (Gustafsson, 2009, p. 93). Fox, et al. observed cases of incompatibility even within standards themselves. For example, HL7 (Health Level 7) Version 3, a widely used set of standards for managing and sharing clinical records, is itself not fully compatible with its previous version (Fox, Sahay, and Hauswirth, 2009, p. 131).

PHIS will eventually link with electronic health records (EHRs) from hospitals; healthcare providers, including PhilHealth (Philippine Health Insurance Corp., a government-operated corporation providing universal health care service for Filipinos); pharmacies and laboratories. Connecting to these fragmented systems will give PHIS a more comprehensive, more representative way for aggregating and disaggregating health data on national and subnational levels. It will also help resolve the issues of duplication of data, encoding efforts and errors when gathering and processing health metrics data.

If PHIS can link directly with hospital patient records, insurance claims and laboratory records, for example, the system can make estimates that are more granular and closer to the population. This richer data set, however, will still have problems of coding errors, double-counting and non-standard coding. Example, one patient may be counted as several patients, owing to her case being recorded in the hospital patient records, the insurance company and the hospital billing records.

4. SOLUTIONS

The critical issues cited in the previous section can be resolved with existing and emerging technologies. The basic building blocks for PHIS are eXtensible Markup Language (XML), enterprise architecture (EA) and service oriented architecture (SOA) -- they provide a framework for creating an interoperable, versatile HIS. They also embody a set of guidelines and good practices to

facilitate governance and security (Gorton, 2006; Newcomer and Lomow, 2005). On the other hand, the emerging approach to bridge the semantic gap is through the use of ontologies (Fox, Sahay, and Hauswirth, 2009; Garcia-Sanchez et al, 2008). Note that for this paper, the terms application, software, features and services are used synonymously.

4.1 XML and Data Independence

As discussed earlier, interoperability problems mostly arose from data that are tightly bound with specific applications. The eXtensible Markup Language (XML) is a way to format data that is locked into specific applications. XML stores data as text files and not in binary format, making it readable by any application. XML describes data through tags similar to HTML (hypertext markup language, the coding standard that makes the world-wide web possible). The XML standard is open, free, simple and portable. Through XML, developers, vendors and clients can agree on standard ways to store and exchange data (Newcomer and Lomow, 2005, p. 3).

Because XML is an open standard, software vendors and developers now use it as the way to universally exchange data. Oracle, MS-SQL and even Access and Excel can export data into XML. Because of this, porting to XML these days is a trivial project. Because of its availability and wide acceptance, XML has become one of the foundations of a suite of integration solutions called web services (Gorton, 2006).

4.2 Enterprise Architecture (EA)

Gorton (2006), defines architecture as a system's structure, its components and their respective functions, and how these components communicate. Gorton also states that architecture must address non-functional requirement issues like quality, and technical and business constraints (such as governance and privacy). Archi-

ture can refer to a wide range of contexts in IT, from a macroscopic framework -- the blueprint for the IT system of a whole organization -- to a design for one particular application. For this paper, architecture refers to the high level abstraction of PHIS.

Stansfield, Orobato, Lubinski, Uggowitz, and Mwanyika (2008) recommend using EA to define the macroscopic framework for a national HIS. According to Stansfield, et al., "The enterprise architecture provides the missing link to guide development and implementation of national health information systems." (p. 7) An overview of EA is needed to understand why it is the recommended approach.

EA places software development against the larger context of enterprise -- the whole business organization and how each individual part works with others. Ross, Weill and Robertson (2006, p. 51) identify four common elements in EA: core business processes, shared data that drives these core processes, key technologies for interoperability, and key users (or customers).

When defining software features and requirements, EA always refers back to the needs of the whole enterprise, in the context of the core processes and key customers. The architect works closely with the clients to define core business processes and users in the entire organization. Then they discover how existing (or non-existing) applications contribute (or will contribute) to those business processes. Then, EA identifies cross-functional (or horizontal) applications to support common processes like email, data storage and authentication, and line of business (LOB, or vertical) applications to support processes like accounting, HR and marketing (Bernard, 2005).

Stansfield et al. have provided a preliminary list of business domains, processes and users of a health information system. Their table is reproduced, below.

Table 1. Starting point for identifying business domains, processes and users for PHIS.
This table may be used as a guide for initial scoping of features for PHIS (Source: Stansfield, et al., 2008)

Business Domain	Business Processes	Archetypical Users
Health Services	Patient registry Individual health record Registration of death Registration of birth Classification of disease Classification of symptoms Classification of procedures Notification of reportable diseases	Patient/guardian/parent Chief health officer Physician Community health worker Trained birth attendant MCH worker District health manager Director of primary health care
Laboratory	Collect and register specimen Determination of results Associate result to patient Notification of reportable diseases Classification of disease	Chief health officer Physician Surveillance officer Laboratory technician
Pharmacy	Central stock registration Facility stock registration Supply chain & distribution Patient registry Classification of disease Treatment plan and prescription	Chief health officer Physician District health manager Provincial health manager Pharmacist Central Stores manager
Human Resources in Health	Taxonomy of health workforce Recruitment, credentialing, hiring of health workers Monitoring deployed workforce Reporting priorities for recruitment & training	National health manager National finance manager Provincial health manager District health manager Facility health manager
Environmental Monitoring	Water quality and access mapping Sanitation resources and access Environmental conditions & history of natural disasters & events Classification of monitoring procedures Routine environmental monitoring	Chief health officer Physician District health manager Provincial health manager National surveillance officer
Decision Support	Family of International Classifications Access to health protocols & research Aggregation of routine data Linking of routine and population data Budget & expenditure reporting Analysis and representation of data Monitoring of urgent health events MDG and M&E reporting	Chief health officer Physician District medical officer Provincial medical officer Global M&E officer Community health worker
Finance	Patient services fee for service collection Health insurance enrollment Health services insurance settlement National and sub-national budgeting National and sub-national expenditure tracking National and sub-national revenue tracking	Chief health officer District health manager Provincial health manager National health finance officer National treasury finance officer

The table reveals recurring themes. Registration, categorization and aggregation of data appear to be common business processes. We can also identify possible shared data, like patient demographics, medical conditions and procedures, births, deaths and causes of deaths. This could further be investigated by the architect and development team designing PHIS.

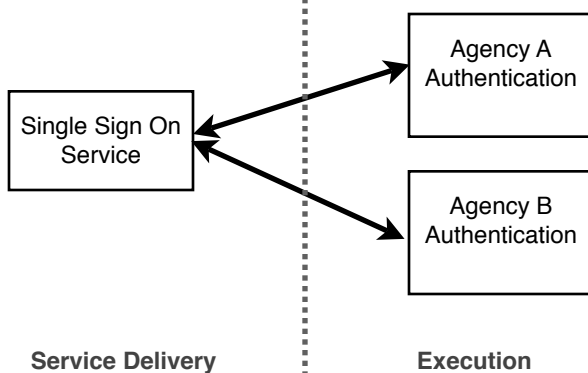
4.3 Service Oriented Architecture (SOA)

EA provides a starting point to describe PHIS in broad strokes, but to draw the technical specifications for interoperability, PHIS must turn to service oriented architecture. SOA refocuses software development by viewing applications as a set services that exchange standard messages.

Gorton (2006) defines services as "...autonomous independent applications, not classes or components that are tightly bound into client applications." Services only need to know how to communicate (exchange messages) with other services, hence the service and the messages they exchange are the only dependencies in SOA. Services do not need to know anything about specific applications or platforms. This simplicity makes services flexible and extensible. Owners of services can easily change service implementation without affecting consumers of the services (Gorton, pp. 222-223).

Granting access permissions by different agencies is an example of an issue solved by separating service delivery from execution. The different organizations participating in PHIS have different systems for authenticating their users, a recipe for user confusion and frustration. PHIS could create a single sign-on (SSO) *service* that sends simple messages to the other systems: is this user allowed to use the system or not? If so, what is the user allowed to do? Carrying out the authentication procedure is delegated to the agency's system and is not performed at the PHIS level. The different agency systems need only respond to those two questions (Illustration 1).

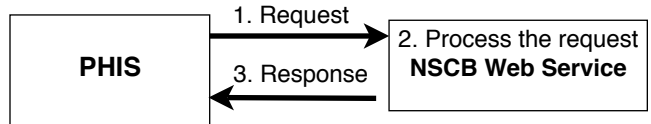
Illustration 1. A service-oriented view of the flight booking process.



One of the source agencies for PHIS is the National Statistical Coordination Board (NSCB). As mentioned earlier, the NSCB data is stored in a Microsoft proprietary platform while PHIS may be built on an open source platform. To facilitate data exchange between the NSCB website and PHIS, NSCB can be turned into a web service that delivers information requested by any entity

(human or software) like PHIS. The NSCB service has three basic functions: (1) accept a message containing a request, (2) *attempt* to process the request and (3) deliver the response through another message.

Illustration 2. Service-oriented approach. NSCB is a component that provides a service to PHIS.



Step 2, the attempt to process the request, makes PHIS a flexible system. If NSCB Web could not execute the request, it passes the data to another service or responds with an appropriate error message. Thus the processing of the request is decoupled from its execution, making data independent from the application. This facilitates the ability of the system to delegate tasks to other applications, making the execution of services independent of any particular application or implementation. This decoupling bridges the gap between business users and technology developers. It lets business users describe their desired application in terms of business services -- a vocabulary that is close to their domain. At the same time, SOA also allows technology developers to translate these service descriptions into technical solutions (Newcomer and Lomow, 2005, p. xxvi).

Combined with web services, SOA makes it possible for different systems to interoperate at relatively lower costs than if organizations had to completely rewrite code or change applications (Gorton, 2006, pp. 217-222). In SOA, services work together by exchanging messages formatted in XML. SOAP is the standard way of structuring messages using XML. Through SOAP, different applications can exchange data and execution requests over a network, *regardless of platform* (Newcomer and Lomow, 2005, p. 20.)

Gorton (2006, p. 225-226) describes four basic functions provided by web services, all enabled by XML:

- Service discovery - a registry of web services, allows other services to easily find other suitable services (UDDI or Universal Description, Discovery, and Integration)
- Service description - the standard way to describe what a service does (WSDL or Web Service Description Language)
- Remote service requests - a way to ask a service to execute a request (SOAP or Simple Object Access Protocol)
- Security and reliability - (WS-* standards; these are other web service standards that are being developed)

In addition, SOA includes business process modeling (BPM), a higher level of abstraction that allows non-technical business analysts to define services that resemble the actual business processes. BPM improves governance by engagement. Non-technical stakeholders can design services without knowing how to program. BPM also improves flexibility. When business process changes, the application can be remodeled based on the new proc-

esses. SOA and its suite of tools and solutions will enable PHIS to interoperate with other systems, regardless of platform, in a secure manner over the Internet at a lower cost and less effort.

4.3.1 PHIS Services

The Framework and Standards for Country Health Information Systems (Health Metrics Network, p. 18), divides the components of an HIS into the IPO method: Inputs (resources needed to run an HIS), Processes (standard indicators), Outputs (the information products and their dissemination to stakeholders).

The list of PHIS core processes may also be categorized into Input, Processing and Output. Input includes processes that allow entities (human, software or devices) to enter data into PHIS. Processing encompasses statistical analysis and knowledge discovery. Output involves processes for formatting and disseminating the information from PHIS. Newcomer and Lomow (2005, pp. 138-140) proposed to classify services as atomic (basic services that perform only one function) and composite (made of other services) -- a clearer way to classify services. From Newcomer and Lomow's classification, these are possible PHIS atomic services:

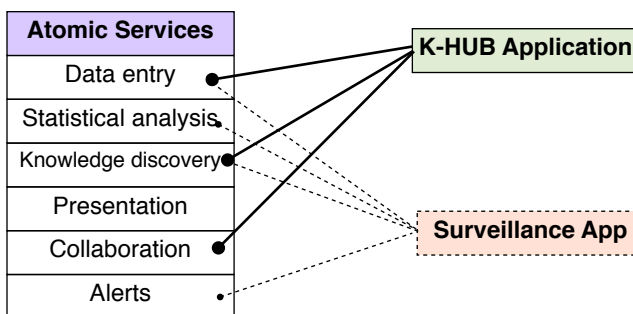
- Data services: entering (and coding), storing and retrieving data
- Data analysis: statistical analysis, knowledge discovery (ie, data mining)
- Presentation: formatting, graphing, charting, visualization
- Communication and collaboration: alerts/notifications, email, contact management, egroups, blogs, forums, etc.
- Policies: security, access policies, service level agreements

Composite services are those formed by combining other services. They would allow PHIS to be a flexible information system. For example, the issue of creating reports for policymakers versus community health workers can be resolved by combining different services in a plug-and-play approach. K-HUB, a system envisioned to be the knowledge management (KM) hub of DOH, can also be built by reusing atomic and composite services. Composite services for PHIS include:

- Monitoring and Surveillance
- Outbreak report and detection
- Web publishing/content management
- Patient records management
- Monitoring and deploying health workers
- Knowledge management for planners, leaders and decision-makers
- Resource budgeting/allocation
- Disaster response
- Decision and policy support

For instance, K-HUB may be assembled from services like data entry, knowledge discovery and collaboration services. K-HUB shares common services with a surveillance system, such as data entry, but may also combine other services like statistical analysis and alerts.

Illustration 3. Two different applications (K-HUB and Surveillance) can share a common set of atomic services. K-HUB and Surveillance are composite services.

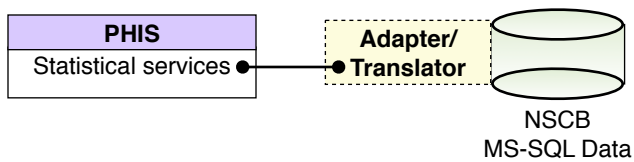


4.3.2 Interoperating with Existing Applications

The following is a simplified overview of how legacy applications can be modified to behave like services, at relatively low effort and cost. The terminologies and concepts are more layered and intricate in the real-world usage.

The key to interoperating with legacy applications is by developing adapters to translate between the application and the service oriented system. In everyday life, a major problem for travelers is the incompatibility of electric outlets in different countries. To solve this, it is cheaper and easier to buy a socket adapter than to ask nations to change their existing outlets. Data in NSCB may be stored in MS SQLServer (MS-SQL). The ASP-based website acts as an adapter that allows web users to extract the MS-SQL data from NSCB. It is also possible to extend this web-based adapter by using XML tags to format, map and transform data using standards defined by SOA. In this way, NSCB can participate in PHIS without changing its database structure or undertaking major code rewrites. NSCB developers can build an adapter using Java and XML that would allow the proprietary NSCB database to act as a PHIS service, thus opening it for information exchange.

Illustration 4. By building an adapter, the proprietary NSCB database is modified to behave like a PHIS service. This allows the NSCB database to participate in PHIS.



In addition to web services and adapters, there are other components of SOA that extend its capability to connect with legacy systems and new systems. This paper will provide only a very brief overview of how this is done. Sartipi, Najafi, and Kazemzadeh (2008, p. 171) identify four main elements of SOA: the services themselves, their front-end applications, service repositories, and the service bus:

- Front-end applications - the software used at the point of service delivery (ie, by physicians, laboratories, pharmacies, etc.). These can be composed by combining different core services.
- Repositories - are registries containing descriptions of services, what messages they accept, in what format and the results they deliver.
- Enterprise service bus (ESB) - acts as a bridge between all services, keeping repositories of available services and how these services can be invoked. By providing a middle layer, the ESB reduces the load over the system. Instead of scanning the entire system and working directly with the services, an application only needs to subscribe to the ESB. The ESB also allows better handling of service disruptions, since services publish their availability through the ESB (Newcomer and Lomow, 2005, p. 195).

The full suite of SOA and web services enable PHIS to be a plug and play environment. By complying with SOA, other systems, whether existing or new, can work with PHIS. In this way, SOA facilitates scalable development and implementation.

4.4 Project Governance

PHIS governance consists of two elements: how the project will be governed by the different partners, owners of data sources and end users (hereafter, “project governance”); and the set of policies that governs the service oriented HIS (hereafter, “service policies”). Project governance is discussed in this sub-section while service policies are discussed in the next.

There are different EA governance approaches towards developing and maintaining enterprise information systems, but they have common points described in this section. The first task is to assign an enterprise architect who will be responsible for the high level design of PHIS. Newcomer and Lomow (2005, p. 6) adds that the architect also “[o]versees the development of reusable services, identifies a means to store, manage and retrieve service descriptions when and where they are needed.” DOH may either select someone from IMS (Information Management Services) and train this person; hire outside expertise, or do a combination of both (hire outside expertise who will work with an understudy from IMS).

Ross, Weill and Robertson (2006) state that “the key element of enterprise architecture is the set of standardized processes and

shared data built on its single instance...” (pp. 48-49) The architect must therefore lead the way to identify these core processes and data by engaging major stakeholders -- source agencies and target users. The architect works with data owners and PHIS target customers to identify existing applications that may be transformed into reusable, shared services.

Engaging stakeholders in a transparent process as early as during the conceptualization and all throughout the development of PHIS is crucial. Its importance is emphasized by the Health Metrics Network (2007, p. 14), Ross, Weill and Robertson (2006, pp. 135-141), and experts like Dr. Alvin Marcelo (online interview, 18 October 2009). Dr. Marcelo reiterates the need to engage both public and private sector partners. Consulting the stakeholders helps disseminate awareness and improves buy-in on the project. The consequence of skipping will be scope and feature creep and lack of buy-in from the intended users and data source owners.

The architect shall convene, and report, to the Governance Committee (GovCom) composed of representatives from PHIN, including subject matter experts on IT, health metrics, and other relevant domains. The architect drafts an EA management plan (or the master plan) describing PHIS in broad strokes and submits this for review by the GovCom. This master plan is reviewed annually and revised according to need (Bernard, 2005, pp. 83-85).

The GovCom will also take charge of defining key documents, security policies, technical specifications and expected responsibilities and accountabilities of data source owners. A major challenge for the committee is to map out a strategy to ensure continuous use of PHIS. This task includes thinking of strategies so users and data owners regularly use and update the system; and ensuring adequate funds to shoulder the costs of modifying legacy systems to interoperate with PHIS.

Note that in the recommendation for annual review and modification of the master plan, EA exhibits a basic design principle -- to accept the reality that business needs evolve through time. Part of EA design is to make information systems flexible enough to evolve with the changing business needs.

4.5 Service Policies

Service policies, or simply “policies” include access rights (who or what applications have access to which data and services) and service level agreements (service reliability, performance, cost, etc) (Newcomer and Lomow, 2005, p. 58). Some policies (like service reliability) are more likely to be non-functional requirements and do not get captured as features in an application. SOA provides a way to create and enforce contracts between services by representing these policy rules as a “collection of machine-readable statements” (Gorton, 2006, p. 224).

In tightly coupled applications, policies are often embedded in the system. Thus, user permissions from different organizations could not be easily imported or mapped into other information systems. The traditional solution -- create new user accounts and define their rights in each system -- resulted in more effort and user frustration in having to keep different accounts and passwords per system. In the context of PHIS, which will involve users and applications from different systems, the administrative burden of creating and maintaining accounts per new participant will be great.

In SOA, policies can be embodied as contracts contained within services, or kept in a separate repository of policies. By using contracts, PHIS could enforce these policies by requiring its services to consult the policy engine. The services then automatically

negotiate with each other, based on the service contracts enforced by the policy repository.

4.6 Privacy, Confidentiality and Technology

Win and Fulcher (2007) point out that privacy and confidentiality are a vital concern to a system. Patients and healthcare providers, for example, may oppose integration of records and databases if they perceive it as a threat to privacy. The authors discuss two general models. General Consent with Specific Denials is patterned after an opt-out model -- the assumption is that of general consent to use, unless the patient denies it. General Denial with Specific Consent Models is an opt-in model. It is more restrictive since it assumes no consent unless granted. This second model was seen to hinder work, especially during emergencies (pp. 91-93).

The issue of privacy is compounded more because it is intricately related with data integrity and security. To avoid duplication of patient records, each citizen could be assigned a unique identification code (UID). In the Philippines, however, giving everyone a UID is a contentious issue. Having been under a dictatorship for more than twenty years, opposition political parties and activists have strongly resisted a national identification system that could have become the basis for UID (Burgonio, 2008; Napallacan, 2008).

Interestingly, consent in some of the Australian pilot projects were enabled by security technology, including digital signatures and iKey (a USB device containing a unique signature that can be unlocked by the patient's PIN). The Philippines may look into this as a workaround over using a traditional and controversial ID system.

Countries and federations like those in the European Union (EU) have successfully implemented UID systems (Hickley, 2008); their example may help promote the justification behind the UID. UK issues the National Health Service number (NHS) which is unique to every patient (Rashbash and Newton, 2006, pp. 141-143). Government may encounter less resistance if it used a national health service number through DOH or PhilHealth, instead of trying to enforce a national ID system.

Realizing the importance of privacy trade-offs in furtherance of education, UK also implemented some access policies that could serve as a model for the Philippines. In the UK, cancer registries (and other selected registries) are open to academic research with some caveats. This policy is reviewed annually by government to gauge the effectiveness of the trade-off (Rashbash and Newton, 2006, pp. 143).

Data linking or data matching may provide a workaround to disparate, anonymous datasets. Data linking/matching is accomplished through a combination of statistics and creative techniques akin to jigsaw puzzle solving. Lyons, et al. (2009) demonstrated the use of statistical probabilities on 500 million hospital records to link disparate records including social security, hospital databases and the National Health Service Administrative Register (NHAR). Using various techniques, the authors achieved different levels of data matching, the highest being 95.2% matched.

Another technological workaround is to use data mining to find patterns in record fields that could reveal identity, and then add random noise to these fields to preserve anonymity without sacrificing the integrity of data analysis (Atzori, Bonchi, Giannotti, and Pedreschi, 2006; Islam and Brankovic, 2004).

4.7 Ontologies

As discussed in Subsection 4.4 (Semantic Interoperability), different standards in encoding and transmitting data have led to the semantic gap problem. Although SOA addresses technical interoperability problems, the semantic gap is something it does not specifically address. (Fox, et al., 2009, pp. 130-137). This subsection looks at emerging technological solutions that help resolve two important problems in all information systems: the translation problem (or the inability to map one data set into another without human intervention), and the data evolution problem (how to easily modify databases to reflect changes in business requirements).

An ontology is a way of representing knowledge. It consists of a hierarchy of concepts (or taxonomy), and their relationships and the logical rules governing these relations (Passin, 2004, p. 15; Daconta, Obrst, and Smith, 2007, p. 185; Gorton, 2006, p. 244). An ontological diagram bears a resemblance to a database entity-relationship diagram (ERD). Ontologies are machine-readable and aid computers to automatically map one coding standard into another. With the help of a reasoning engine, an ontology can make inferences about the contents of unstructured documents like email, medical transcriptions and web pages, for example.

The two statements, below, illustrate the problem of representation and the role of ontology.

- A. Dr. Bengzon treated Juan Santos of pulmonary tuberculosis.
- B. Juan Santos filed an insurance claim for TB.

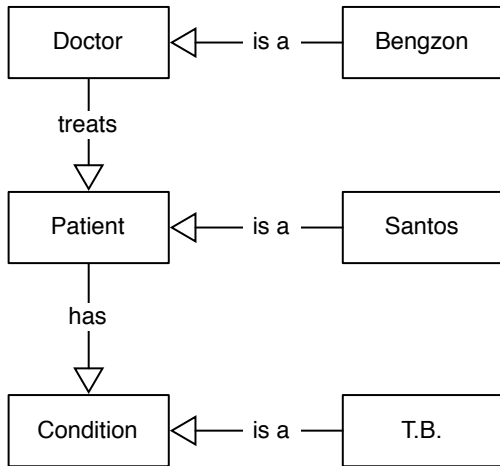
Humans will at once infer that (a) Dr. Bengzon is the doctor of Juan Santos (relationship), (b) pulmonary tuberculosis and TB are the same (synonyms), and (c) Juan Santos cannot treat Dr. Bengzon (restrictions). Computers have no way of inferring these concepts automatically. If statement A is a clinical record and B is an insurance record, a program may count the cases as two separate incidents of TB.

To solve this, applications reading patient records can consult an ontology. The ontology could look like a series of descriptions:

- A. Bengzon is a doctor.
- B. Juan Santos is a patient.
- C. Doctor treats patient.
- D. Patient cannot treat doctor.

These relationships and restrictions are better seen through an ontology map:

Illustration 4. A simple ontological model representing doctors, patients and illnesses.



Such a representation can be standardized into a coding system that makes the relationships machine readable. With less coding, different ontologies (like medical vocabularies in different languages) can therefore be mapped into one another. OWL (Web Ontology Language) is one of the emerging standards for modeling ontologies. It is built on RDF or Resource Definition Framework, itself built on XML. OWL has three versions: Lite, DL and Full. (Passin, 2004, p. 22 and 164).

Below is an excerpt from Gustaffson's thesis (2009, p. 189) showing a portion of a medical ontology in OWL. This particular snippet is a list of values that are accepted in the system.

```

Namespace(rdf =
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Namespace(rdfs =
<http://www.w3.org/2000/01/rdf-schema#>)
Namespace(owl =
<http://www.w3.org/2002/07/owl#>)
Namespace(somwebValueList =
<http://www.somweb.se/ontologies/somwebValueList.owl#>)
Ontology(
<http://www.somweb.se/ontologies/somwebValueList.owl>
Annotation(rdfs:comment "Value list
ontology of the
SOMWeb community.")
Class(somwebValueList:Drug
annotation(dc:creator "termValues")
annotation(somweb:medviewType multiple))
Class(somwebValueList:Occup
annotation(dc:creator "termValues")
annotation(somweb:medviewType regular))
Class(somwebValueList:ResultNow)
annotation(dc:creator "termValues")
annotation(somweb:medviewType regular))
Individual(somwebValueList:Eucardic
type(somwebValueList:Drug)
annotation(dc:creator "termValues")
annotation(rdfs:label "Eucardic" @sv))

```

```

Individual(somwebValueList:Doktacillin
type(somwebValueList:Drug)
annotation(dc:creator "termValues")
annotation(rdfs:label "Doktacillin" @sv))
Individual(somwebValueList:Inredningssnickare
are
type(somwebValueList:Occup)
annotation(dc:creator "termValues")
annotation(rdfs:label "Inredningssnickare"
@sv))
Individual(somwebValueList:helt_utlakt
type(somwebValueList:ResultNow)
annotation(dc:creator "termValues")
annotation(rdfs:label "helt utläkt" @sv))
)
)

```

Ontologies serve as translation maps that allow different systems to automatically exchange data that are coded in different standards. The following subsections discuss some cases that successfully used ontologies for their health information projects.

4.8 Case Studies

The following are just a few of the interesting projects and studies reported in various papers and publications. They are described here to show the various technological possibilities for PHIS. It is possible to study these projects (by downloading or using the demos available online) in preparation for PHIS. Some may even serve as foundations for SOA components for PHIS. The appendix contains links to the resources and technologies cited by the studies.

4.8.1 PPEPR

Fox, et al. (2009), created an extension called semantic SOA (sSOA) to address the semantic gap problem caused by the incompatible versions of HL7 Version 3 and HL7 Version 2. The team developed PPEPR (Plug and Play Electronic Patient Records), an application to demonstrate how sSOA bridges the semantic gap while entailing relatively less effort and costs. Before PPEPR, integrating disparate HL7-based systems entailed manual code rewriting, entailing expensive work and resulting to maintenance problems.

PPEPR provides an automated way to map one standard to another (in this case, the different versions of HL7), facilitating the integration of systems that use different coding standards. First, the service maps the disparate standards on its own, using ontologies. Next, it requests human approval to validate the translations. The point of PPEPR is that it was able to make intelligent guesses on how to translate disparate data, which eases the effort needed on the human at the validation point.

To build PPEPR, the team used the following standard web service technologies:

- WSMX - the web service execution environment
- WSML - the web service modeling language
- WSMT - the web service modeling toolkit
- WSMO - the web service modeling ontology

It is interesting to note that, using SOA, the team was able to make a non-web service compliant standard (HL7 Version 2) behave like a service and communicate with the web service compliant HL7 Version 3. Also, since PPEPR is built as a service

based on SOA design principles, it can stand on its own or act as a pluggable service into existing EHRs. Because the mapping is delegated into a separate layer involving ontologies, it is possible to reuse PPEPR with different standards by modifying the ontologies involved.

Fox, et al. also contained initial metrics on how long it took to achieve the integration. It took the team an average of 1.5 days to create HL7 ontology models, transformation rules (in XSLT) and mapping definitions. At run time, it took PPEPR 2-3 seconds to exchange messages on a broadband connection. These 2-3 seconds included the time to automatically consult the ontological mappings and transform the messages. (Note that PPEPR is not open source).

4.8.2 SOMWeb

The **SOMWeb** (Swedish Oral Medicine Web) project was a knowledge management project developed by Gustaffson (2009) in Java as part of her PhD thesis. The goal was to support a dentists' community of practice (CoP), by providing an electronic way for knowledge sharing. SOMWeb provided basic functionality needed to spur collaboration among members of the CoP: a way to input cases, a case browser that allowed doctors to review related cases, and a way to discuss the cases collaboratively. Gustaffson experienced difficulty in enticing the doctors to use the application. However, when they did use it, doctors reported benefiting from SOMWeb, some even saying they learned more from it than attending conferences (p. 52).

Gustaffson used OWL DL to model the ontologies of oral medicine and Jena (an open source semantic framework for Java, developed by HP) to manipulate the ontologies. In the thesis, she reported lessons learned from developing SOMWeb with OWL and RDF, among them:

- Using OWL and RDF made it easier to update the data as requirements changed. The ontologies also improved the relevance of cases displayed in the case browser.
- Analysis of the SOMWeb usage logs showed that it successfully facilitated and enhanced distributed collaboration and distance consultations (an outcome relevant to the Philippine setting). For instance, logs showed that doctors often reviewed cases collectively during meetings. SOMWeb's email feature has also shown increased contact among CoP members *outside of meetings*.
- OWL is still a relatively new technology and developers need time to understand and use it.
- Since it is a young field, there is a lack of reusable ontologies. This lesson is also particularly important in the Philippine context. The Philippines has more than 100 languages (Linguistic Society of the Philippines Website), each with its own way to describe medical conditions. Developing Filipino ontologies is going to be a major, but attainable effort for PHIS. One strategy for this is to encourage and sponsor interdisciplinary research in medical, linguistic and IT fields.
- OWL DL is limited in its capacity to describe integrity constraints, which impacted on the flexibility of the application. OWL Full allows representation of integrity constraints, but SOMWeb architecture was built to use only OWL DL. Since SOMWeb was not designed from the SOA framework, it was difficult to make the change from DL version to Full.

4.8.3 Mirth

Mirth started as an open source solution to integrate different versions of HL7. It is now called Mirth Connect, to distinguish it from other support products that have resulted from its success -- Mirth Results, a service for handling clinical data (including patient demographics and laboratory results), and Mirth Match, a plug-in that handles patient identity and record reconciliation. Mirth Connect is a semi-automated integration engine that allows humans to filter messages for information and map these into a standard coding system. Mirth Connect stands in the middle of different systems: patient records, laboratory results, hospital and billing records, and facilitates the routing of information from one system to another.

The Mirth interface uses a web browser, presenting information in web forms that are familiar to most users who have surfed the web. Mirth is built on a client-server framework with an enterprise service bus (ESB) similar to SOA. It is not fully SOA, but discussing that is not within the scope of this paper. Mirth is based on European interoperability standards defined by SemanticHEALTH and RIDE (Bortis, 2008).

4.8.4 Google Health

Google Health is a free records management system developed from Google's own implementation of web services. It allows users to store vital health information like immunizations, allergies, conditions and demographics. It lets users define security levels, upload documents and import medical records. Google Health users can allow authorized health care providers to access their personal records and third party developers and institutions can connect to and use Google Health through the free Google Health APIs.

Google Health is an example of an application built over a service-oriented framework. Its APIs are services that provide responses and requests sent as standard messages through the web. Developers do not need to know the specific implementation of Google Health to connect their services to it. The advantage of Google Health is that it is freely available and runs on Google's reliable computing cloud services. It also is already running, hence, organizations can simply connect their applications to ride on the services it already offers. The disadvantage is that it is not totally open source -- the client libraries are released as open source but the core code is not. Hence there is less control of its functionalities. Another disadvantage is the control of the data repository is not within the organization.

4.8.5 Data Evolution with Mesodata

Mesodata is an additional layer in a database, describing extra information about domain structure, additional relations, and possible operations on the data (deVries and Roddick, 2004, p. 429). Working in a similar manner as ontologies, mesodata are a bridge that help databases to evolve gracefully with changing business requirements. deVries and Roddick sorted database changes into three possible categories:

- Attribute representation - eg., if the identification code changes from alphanumeric to integer.
- Domain constraint - eg., a field for landline phone numbers (7 digits) is expanded to include mobile numbers (11 digits).
- Meaning change - eg., San Fernando municipality is promoted to a city and is detached administratively from the province of Pampanga. (p. 430)

When any of these three changes happen in a traditional database, administrators adjust by writing code to execute strict (convert everything), lazy (convert only data that is requested) or no conversion. The best option is to convert everything, but this entails a lot of cost and effort, and hinders a smoother data evolution.

deVries and Roddick created a prototype of a mesodata system using MySQL database and Java wrappers to handle the mesodata relations. The authors showed that mesodata helped “reduce or remove the necessity of schema conversion, schema integration, data conversion and application change as well as maintain or expand the schema’s information capacity.” (p. 439)

5. CONCLUSION AND RECOMMENDATIONS

PHIS will bring positive outcomes that could help improve the health of Filipinos. However, developing it is expected to be a challenging, long-term undertaking. The major challenge faced by PHIS is technical interoperability, which can be achieved by designing it based on the principles of enterprise architecture and service oriented architecture.

Due to lack of time and space, this paper excludes other interesting case studies and technologies useful to PHIS. For example:

- Mobile and other devices. SOA provides the framework to facilitate interoperability with mobile devices like cell phones and PDAs (Newcomer and Lomow, 2005, p. 379-381).
- Social data mining. Usage logs in PHIS, especially in K-HUB can be studied using text mining and social data mining to find patterns, clusters and associations among the experts and the information (Ochoa, et al., 2008, pp. 291-320). This could in turn strengthen PHIS by value-added features such as content recommendation and improvement of ontologies.
- Data mining applied on EHRs: data mining on health records can serve as the backdrop for a hypothesis engine (Shillabeer, 2008) that could be used for disease detection and can even be extended to early detection of disease outbreaks. Decision trees and rules may also be used to generate clinical guidelines for diagnoses (Sartipi, 2008).

Guidelines and Recommendations for PHIS development. The following guidelines are based on a combination of principles from SOA, software development methodology, public health and change management, and are intended to improve the success rate of PHIS implementation.

1. Appoint an architect, convene a Governance Committee and define the master plan. The architect shall draft the master plan in close consultation with the Governance Committee which represents the policy and decision making body of the PHIS project (Details in Section 5.6, [Project Governance](#)).
2. Train project staff and key stakeholders (perhaps the GovCom) to give an overview of enterprise architecture and service oriented architecture, and how these will be used to design and develop PHIS.
3. Develop with end-use in mind. How will users use the information from the applications? What do they really need? Develop applications in the context of serving these needs. Gather only data that is relevant to end-use. It is better to focus resources on gathering data and creating applications that will be used immediately or in the near future (Mo-

stashari, Tripathi, and Kendall, 2009, p. 355; Health Metrics Information Network, 2007, p. 13)

4. Treat users as co-developers. Developers need to include non-technical end-users even during the conceptualization and design stages and all throughout the development phase (usually through user testing). This allows for a system that is more responsive to the needs of users. *Engage users/stakeholders from the start*. It is important for PHIS to involve everyone who will be contributing to and using the system. Although DOH plans to start with the public sector first, it is still paramount to involve the private sector too. The reason for early engagement is to improve buy-in from everyone. Bringing stakeholders late in the process lowers the chance of acceptance (Health Metrics Information Network, 2007, p. 13; Ross, Weill and Robertson, 2006, pp. 135-141).
5. Establish a legal foundation for privacy and confidentiality policies. This includes creating policies and laws not just to protect privacy but also to allow informed use of health information for activities. For example, overriding the privacy policies may be needed for academic research (provided the data is treated with confidentiality) or during emergency situations, when doctors, for example, may need to ask a system administrator to override privacy policies to save lives (Win and Fulcher, 2007, pp. 94). Also look into the use of electronic security devices like iKeys as workaround to the strong resistance against a national ID system.
6. Applications as services; separate service delivery from implementation. Identify core business processes (eg, the process of gathering information), then find applications (existing or soon-to-be-developed) that could provide services to support these processes. Think of data-gathering by different users as a process that could be served by a standard data service -- a software component that allows any user to input, store and retrieve data. These services may then be re-used by other entities or organizations. (Discussed in 5.3, [Service Oriented Architecture](#))
7. Use open standards. This encourages flexibility, interoperability and minimizes being locked into vendor-specific solutions. (Bortis, 2008, p. 652; Gorton, 2006, p. 218).
8. Take small steps. Tackle small projects with high chances of success. SOA’s modular framework allows application development in a series of small steps. Example, start by developing a service to bridge data interchange between the DOH website and the NSCB website. Starting small has the benefits of achieving early success which in turn builds more confidence and buy-in from everyone. (Ross, Weill and Robertson, 2006, p. 118)
9. Evolve gracefully. Instead of radically changing everything to a service-oriented infrastructure, start by transforming existing applications into services. Since service requesters do not need to know how a service is executed, developers can start by rewriting a few service providers, without impact on the requesters (Newcomer and Lomow, 2005, p. 92).

6. ACKNOWLEDGEMENTS

The author thanks Anna Shillabeer, Carnegie Mellon University; Charity Tan, Head of the Knowledge Management Team, Department of Health; and Dr. Alvin Marcelo.

7. REFERENCES

- Atzori, M., Bonchi, F., Giannotti, F., and Pedreschi, D. (2006). Anonymity preserving pattern discovery. *The VLDB Journal* 17(4), 703-727.
- Bernard, S.A. (2005). *An introduction to enterprise architecture* (2nd Ed.). Indiana: AuthorHouse.
- Bortis, G. (2008). Experiences with Mirth: An open source health care integration engine. *ICSE'08*, 10-18 May 2008, 649-652.
- Burgonio, T.J. (2008, January 9). CHR commissioner backs NEDA national ID. *Philippine Daily Inquirer Online*. Retrieved 16 October 2009, from http://newsinfo.inquirer.net/breakingnews/nation/view/20080109-111342/CHR_commissioner_backs_NEDA_nat%92l_ID
- Daconta, M.C., Obrst, Leo J. and Smith, K.T. (2003). *The semantic web: A guide to the future of XML, web services, and Knowledge Management*. Indianapolis: Wiley Publishing.
- Detmer, D (2006). Improving health and health care through informatics. In D. Pencheon, C. Guest, D. Melzer & J.A. Muir Gray (Eds.), *Oxford handbook of public health practice* (2nd Ed.; pp. 466-479). NY: Oxford University Press.
- deVries, D. and Roddick, J.F. (2004). Facilitating database attribute domain evolutions using mesodata. In S. Wang, et al. (Eds.), *ER Workshops 2004, LNCS 3289*, 429-440.
- Fox, R., Sahay R., and Hauswirth M. (2009). PPEPR for enterprise healthcare integration. *eHealth 2008, LNCS 1*, 130-137.
- Garcia-Sanchez, F., Sabucedo, L.A., Martinez-Bejar, R., Rifon, L.A., Valencia-Garcia, R., and Gomez, J.M. (2008). A knowledge technologies-based multi-agent system for eGovernment environments. *SOCASE 2008, LNCS 2006*, 15-30.
- Gorton, I (2006). *Essential software architecture*. Berlin: Springer Verlag.
- Gustaffson, M. (2009). *SOMWeb: Supporting a distributed clinical community of practice using semantic web technologies*. PhD Thesis. Chalmers University of Technology and University of Gothenburg. Retrieved 12 November 2009 from Chalmers Online Library (<http://publications.lib.chalmers.se/records/fulltext/94030.pdf>)
- Health Metrics Network (2007). *Framework and standards for country health information systems* (2nd Ed). Switzerland: Health Metrics Network, World Health Organization.
- Hickley, M. (2008, September 26). British identity cards will be covered in EU symbols. *UK Daily Mail Online*. Retrieved 10 October 2009, from <http://www.dailymail.co.uk/news/article-1061761/British-identity-cards-covered-EU-symbols.html>
- Islam and Brankovic, 2004. *A framework for privacy preserving classification in data mining*. Paper presented at the ACSW Frontiers '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation.
- Linguistic Society of the Philippines Website. Retrieved 12 November from <http://www.lphil.org>
- Lyons, R.A., Jones, K.H., John, G., Brooks, C.J., Verplancke, J.P., Ford, D.V., Brown, G., and Leake, K. The SAIL databank, linking multiple health and social care datasets. *BMC Medical Informatics and Decision Making* 9 (3).
- Napallacan, J. (2008, January 9). IBP Cebu members differ too on national ID plan. *Philippine Daily Inquirer Online*. Retrieved 16 October 2009, from http://newsinfo.inquirer.net/breakingnews/regions/view/20080109-111367/IBP_Cebu_members_differ_too_on_national_ID_plan.
- Ochoa, A., Hernandez, O., Gonzalez, S., Castro, A., Gelbukh, A., Hernandez, A., and Iztebegovic, H. (2008). Social data mining to improve biospired intelligent systems. In E.G. Giannopoulou, *Data mining in medical and biological research* (pp. 291-320). Croatia: In-Teh.
- Mostashari, F., Tripathi, M., and Kendall, M. (2009). A tale of two large community electronic health record extension projects. *Health Affairs* 28 (2), 345-356.
- Newcomer, E. and Lomow, G (2005). *Understanding SOA with web services*. NJ: Pearson Education.
- Passin, T.B. (2004). *Explorer's guide to the semantic web*. Connecticut: Manning Publications.
- Pencheon, D. (2006). Using data and evidence: Introduction. *Oxford handbook of public health practice* (2nd Ed.; pp. 78-79). NY: Oxford University Press.
- Philippine Health Information Network (2007). *Philippine health information system: Review and assessment*. Switzerland: Health Metrics Network, World Health Organization.

Rashbash, J. and Newton, J. (2006). Assessing longer-term health trends: registers. In D. Pencheon, C. Guest, D. Melzer & J.A. Muir Gray (Eds.), *Oxford handbook of public health practice* (2nd Ed.; pp. 140-145). NY: Oxford University Press.

Ross, J.W., Weill, P., and Robertson, D.C. (2006). *Enterprise architecture as strategy: Creating a foundation for business execution*. Boston: Harvard Business Press.

Sartipi, K., Najafi, M., and Kazemzadeh, R.S. (2008). Data and mined-knowledge interoperability in eHealth systems. In E.G. Giannopoulou, *Data mining in medical and biological research* (pp. 159-176). Croatia: In-Teh.

Shillabeer, A. (2008). Building clinical trust in automated knowledge acquisition. In E.G. Giannopoulou, *Data mining in medical and biological research* (pp. 69-106). Croatia: In-Teh.

Shillabeer, A. and Roddick, J.F. (2007). *Establishing a Lineage for Medical Knowledge Discovery*. Paper presented at AusDM'07: Proceedings of the sixth Australasian conference on data mining and analytics, Gold Coast, Australia.

Stansfield, S., Orobato, N., Lubinski, D., Uggowitz, S., and Mwanyika, H. (2008). The case for national health information system architecture. (Unpublished paper from WHO).

Weinberg, J. and Pencheon, D. (2006). Providing data and evidence for practitioners and policy makers. In D. Pencheon, C. Guest, D. Melzer & J.A. Muir Gray (Eds.), *Oxford handbook of public health practice* (2nd Ed.; pp. 194-204). NY: Oxford University Press.

Win, K.T. and Fulcher, J.A. (2007). Consent mechanisms for electronic health record systems: A simple yet unresolved issue. *Journal of Medical Systems*, 31, 91-96. DOI 10.1007/s10916-006-9030-3.

8. APPENDIX: LINKS TO RESOURCES

Products

Google Health is a free health records management system. Google Health APIs are available at <http://code.google.com/apis/health> while the application itself is available at <http://health.google.com>.

Mirth open source products are available at <http://www.mirthproject.org>.

The **Plug and Play Electronic Patient Records (PPEPR)**, developed in Ireland is the project that extends SOA with the semantic ability to automatically translate between HL7 Version 2 with Version 3. Information and a demo is available at <http://www.ppepr.com>

Code and Tools for SOA and Ontologies

ebXML Case Studies that show real-world uses of XML for interoperability are available at http://ebxml.org/case_studies. One case study is on how the US Center for Disease Control and Prevention (CDC) uses it to build messaging services and collaboration protocol agreements.

JBoss is a suite of components that deliver SOA components like ESB, adapters and messaging. JBoss is open source and Java-based. It can be the platform from which an integration project could take off. Available at <http://www.jboss.org>

Jena, a semantic web framework for Java was developed by HP and released as open source. Available at <http://jena.sourceforge.net>

OWL -- Web Ontology Language, Version 2 of the standard developed by the W3 Working Group is described in more detail at <http://www.w3.org/TR/owl-ref>. For a quick overview of OWL, visit <http://www.w3.org/TR/owl2-overview>.

Protege. Protege is a free Java-based tool for building ontologies and may be downloaded at <http://protege.stanford.edu>. The site also contains links to different ontologies coded in OWL and other standards, including those relevant to health, including EHRontology (an ontology for electronic health records), the NCI Thesaurus (created by the National Cancer Institute), Breast-CancerOntology, Cardiology.owl and TAMBIS (biological sciences).

http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library#OWL_ontologies

Standards, Frameworks and Other Resources

Health Metrics Network encourages the development of standardized, integrated national health information systems worldwide. Spearheaded by WHO, it has recommended a framework for standard health metrics and a set of guidelines for creating and sustaining national health information systems. PHIS design and implementation is partly based on the HMN framework. <http://www.who.int/healthmetrics>

Open Clinical contains basic information tailored for multidisciplinary interest in eHealth. It contains updates on various technologies including point-of-care decision support systems and clinical workflow. <http://www.openclinical.org>

RIDE is the EU roadmap to make eHealth systems interoperable. <http://www.srdc.metu.edu.tr/webpage/projects/ride>

SemanticHEALTH is an EU project to solve major problems on semantic interoperability. It is a collaboration of various stakeholders, including WHO and European educational institutions. <http://www.semantichealth.org>

Web service standards are maintained by the W3 Consortium and reside in <http://www.w3.org/2002/ws/>.