November 2011

Ministry of Health & Population

NHSSP

IT Review

providing recommendations
for strengthening HMIS
and piloting HSIS

Strengthening Health Systems-Improving Services

Table of Contents

Table of Contents

List of Acronyms

ADSL	Asymmetric Digital Subscriber Line
ARS	Ayurveda Reporting System
D/PHO	District Public Health Office
DDC	District Development Committee
DHIB	District Health Information Bank
DIC	District Information Centre
DIN	Drug Information Network
DOHS	Department of Health Services
EOC	Emergency Obstetric Care
ETL	Extract, Transform and Load (process)
FCHV	Female Community Health Volunteer
FMIS	Financial Management Information System
GIDC	Government Information Data Centre
GIS	Geographic Information System
GIZ	Deutsche Gesellschaft für Internationale Zusammenarbeit
HIIS	Health Infrastructure Information System
HMIS	Health Management Information System
HSIS	Health Sector Information System
HuRIS	Human Resource Information System
IP	Internet Protocol
ISP	Internet Service Provider
IT	Information Technology
LAN	Local Area Network
LMD	Logistics Management Division

LMIS	Logistics Management Information System
M&E	Monitoring and Evaluation
MIS	Management Information System
MOHP	Ministry of Health and Population
MRD	Master Reference Database
NAT	Network Address Translation
NHIC	National Health Information Centre
NHIPC	National Health Information Policy Committee
NHSP	Nepal Health Sector Programme
NHSSP	Nepal Health Sector Support Programme
OLAP	Online Analytical Processing
PLAMAHS	Planning and Management of Assets in Health Services
SSIS	SQL Server Integration Services
TCP/IP	Transmission Control Protocol/Internet Protocol
TIMS	Training Information Management System
UPS	Uninterruptible Power Supply
WAN	Wide Area Network
XML	Extensible Mark-up Language

Key Definitions

- ***Asymmetric Digital Subscriber Line (ADSL)*** is a data communications technology that enables high-speed data transmission over standard telephone lines. The speed of an ADSL connection is asymmetric, higher toward the customer premises than from it.

- ***Extract, Transform and Load (ETL) processes*** are used to integrate data from multiple sources into a single reporting database. The data are extracted from one or more systems, transformed into a structure facilitating reporting and analysis, and loaded into a dedicated reporting database. There are specialised software products that provide tools to create and manage ETL processes.

- A ***High-Level Architecture*** outlines the overall structure of a complete enterprise system, identifying the major components and how they will interact with each other. Subsequent design activities focus on the details of the individual components.

- An ***Internet Service Provider (ISP)*** is a company that provides Internet access to its customers, e.g. World Link in Nepal.

- The ***Internet Protocol (IP)*** is a global standard dictating how computers on the Internet communicate with each other.

- A computer's ***IP address*** is the Internet equivalent of a phone number.

- A ***Local Area Network (LAN)*** is a private computer network that interconnects computers in a limited area such as a school or office building.

- A ***Wide Area Network (WAN)*** is a telecommunications network that covers a broad area, e.g. connecting offices in different cities. The Internet is a public WAN.

- ***Network Address Translation (NAT)*** is a function provided by a network router that effectively hides the computers on a private network from the rest of the Internet, while still allowing those computers to access the Internet.

- ***Extensible Mark-up Language (XML)*** is a standardised and flexible format used to exchange complex data between computer systems.

1 EXECUTIVE SUMMARY

This assessment was undertaken by Alan Leavy (International Consultant, Options) in collaboration with Deepak Shrestha (Chief, MIS IT, MIS Section, Management Division, DOHS), Dhruba Ghimire (MIS Officer, DOHS) and Pradeep Poudel (M&E Implementation Advisor, NHSSP) from 10-23 October 2011.

Key findings	Recommendations
Software The overall designs of the Health Management Information System (HMIS) and Health Sector Information System (HSIS) pilots are similar, conceptually very simple, and fit for purpose, but the large number of data fields makes the systems complex. Further, poor testing and ineffective contractor and software development management mean that systems are crippled by software bugs, resulting in labour-intensive approaches to work around problems.	• Several quick fixes are possible (e.g. pilot software bug fix initiative, HMIS facility-level data export fix). • The current Management Information System (MIS) Information Technology (IT) Officer could take on an expanded IT and Software Manager role. • A software development methodology to ensure better software quality (e.g. user interface prototyping, user testing during software development process).
Data entry Data are currently entered at central level. In the past reliable Internet connections were not available in most District Public Health Offices (D/PHOs), but ongoing rollout of Asymmetric Digital Subscriber Line (ADSL) Internet services across the country means this has improved dramatically and will only continue to get better.	• Move towards a centralised online system into which facility-level reporting data could be entered directly at the D/PHOs.
Tools and indicators The current HMIS tools and indicators used in HSIS pilot and non-pilot districts are not fully compatible. A consistent set of health service indicators is required from all districts for national annual reporting purposes; differences in tools and indicators mean that data entry is more time-consuming, and result in incomplete data for the pilot districts.	• Review and revise current HMIS tools. • Develop a corresponding HMIS database application in time for the start of the next reporting year. • Use in all pilot and non-pilot districts (with a small set of additional questions/tools for HSIS pilot districts).

Current HSIS pilot vs original HSIS vision The HSIS pilot is a valid first step towards achieving the goals of the original HSIS vision, given the current environment in which it has to operate. However, it has some severe limitations (functionality and scalability).	• Build upon the current HSIS pilot. • Use a phased approach towards developing an integrated information system in line with the HSIS vision. • Use increasingly sophisticated methods for integrating data from multiple systems and a target technical architecture for the overarching system. • A master reference database supporting a uniform coding system is a key component of the architecture, enabling the integration of data from multiple sources. • The District Health Information Bank (DHIB) and its reporting portal are included in the original HSIS strategy, but could be viewed as optional. The functionality they provide is also available directly from the central reporting portal. The DHIB is a replica of the central-level databases albeit containing the data for just one district.
Antivirus protection Computers inspected during the review had Kaspersky Antivirus software installed with up-to-date virus definitions. However, as the antivirus software is not centrally managed, there is no guarantee that this is the case across the board.	• Use the Kaspersky administration tools to provide a central view of all machines on a network: tracking licences, monitoring virus activity, reporting security breaches, and facilitating software and virus definition updates.
Database backup The central HMIS application provides a facility to back up and restore the database; however, it does not work. The backups generated are incomplete and would result in data being lost. Database backups can be, and are being, taken manually, but irregularly and on an ad-hoc basis.	• Create a simple database backup and restore utility, built on SQL Server's native backup functionality. • Use a tool like Windows Task Scheduler to ensure that backups are taken regularly.
System backup There are no system backup tools or processes currently in place. In the case of a general system failure, an application server would have to be rebuilt from scratch. The custom developed applications would pose the biggest problem as they would have to be installed by the original software developers, assuming they still had a copy of the appropriate version of the application. Such failures have already occurred several times leading to system outages of up to several weeks at a time.	• Use server virtualisation to simplify full system backups. • Establish simple and well-documented, manual or automated, backup and restore processes. • A designated person must be responsible for database backup and system backup. This role could be filled by the current IT Officer, MIS section

Central server environment The central HMIS servers are located in a normal office, which is air-conditioned but dusty. Access to this office is not very restricted, opening up possibilities for deliberate or accidental interference with the server hardware by unauthorised personnel. There is a power backup facility, but without enough capacity to guarantee continuity of service during a power outage lasting several hours. However, long power outages are apparently infrequent in the MIS section at central level as it is prioritised during load shedding.	Adopt an incremental approach towards creating a high availability data centre environment, gradually adding backup hardware and eliminating single points of failure: 1. ***MIS server room*** – A small server room could be built within the MIS section, limiting physical access to the hardware and providing an air-conditioned and dust-free environment. Basic server virtualisation, system backup tools, and redundant server hardware should be used to enhance server availability. 2. ***Consolidation of department IT resources*** – There are several systems scattered across the Department of Health Services (DOHS). It would be to the mutual benefit of all divisions to cooperate, sharing or pooling their IT resources to create a more robust hosting infrastructure. 3. ***National Health Information Centre (NHIC)-managed data centre*** – The process of consolidation could eventually lead to the de facto creation of a department- or ministry-wide data centre. Although not yet fully operational, the Government Information Data Centre (GIDC) should be considered as a long-term hosting option, or at the very least as a disaster recovery site[1].
Central network The central servers are not on an isolated and protected Local Area Network (LAN) segment. They do not have a dedicated Internet connection and do not have any priority in accessing the connection shared with the rest of the office. As things stand there is no way to guarantee the performance of online access to the central HMIS system or the planned HSIS reporting portal.	• Establish an isolated LAN segment for the central servers, using a router with good firewall capabilities. • Provide a dedicated Internet connection for the servers and use the office Internet connection as a backup.
D/PHO server environment There were no dedicated server hardware or machine room environments at the D/PHOs visited during this assessment. The HSIS pilot and HSIS applications and databases are installed on user machines.	• Adopting an architecture within which the D/PHO-hosted systems (e.g. the DHIB) are not critical to the continuity of the end-to-end reporting processes. • Using virtualisation to simplify D/PHO technical support and operations, so that locally hosted systems can be brought back online quickly in the event of a failure.

The report also proposes a high-level implementation plan:

[1] A disaster recovery site is a secondary data centre with capacity to provide limited or full IT services in case the primary data centre becomes completely inoperable because of fire, flooding or other such disasters.

2 BACKGROUND AND OBJECTIVES

There are nine Management Information Systems (MISs) within the Ministry of Health and Population (MOHP): the Health Management Information System (HMIS); Logistics Management Information System (LMIS); Financial Management Information System (FMIS); Health Infrastructure Information System (HIIS); Human Resource Information System (HuRIS); Training Information Management System (TIMS); Ayurveda Reporting System (ARS); and Drug Information Network (DIN). In addition, Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) recently supported the development of an additional MIS software: Planning and Management of Assets in Health Services (PLAMAHS).

These MISs are independent systems utilising different coding structures. However, it is proposed to link these systems (with the exception of the DIN) using a uniform coding system under the umbrella of an overarching Health Sector Information System (HSIS). HSIS will not replace the individual MIS; it will provide a platform from which information from all these systems can be linked, accessed and distributed.

This assessment was undertaken by Alan Leavy (International Consultant, Options) in collaboration with Deepak Shrestha (Chief, MIS Information Technology (IT), MIS Section, Management Division, Department of Health Services (DOHS), Dhruba Ghimire (MIS Officer, DOHS) and Pradeep Poudel (Monitoring and Evaluation (M&E) Implementation Advisor, Nepal Health Sector Support Programme (NHSSP). During the assessment the international consultant visited Nepal (10-23 October 2011) and visited the following sites with his collaborators:

- Central MIS section, DOHS (central HMIS system)
- Lalitpur District Public Health Office (D/PHO) (HSIS pilot district)
- Kathmandu D/PHO (HMIS)
- LMIS, Logistics Management Division (LMD), DOHS
- Rupandehi D/PHO (HSIS pilot district)
- Lumbini Zonal Hospital (sentinel surveillance site)
- Dang District Health Office (DHO)(HMIS)
- Rapti Sub-Regional Hospital, Ghorahi, Dang
- Bheri Zonal Hospital, Tulsipur, Dang

The objectives of the assessment were:

- To undertake an IT review of the current HSIS pilot and provide recommendations for immediate improvements and further development to achieve the objectives of the original HSIS strategy.
- To undertake an IT review of HMIS (currently in 72 districts), and provide recommendations for improvement, both short-term fixes and longer-term measures, that complement the development of HSIS.

3 ASSESSMENT

3.1 SOFTWARE QUALITY

3.1.1 Software Design

The individual HMIS and pilot HSIS applications have very similar designs. The functionality they were developed to provide is conceptually very simple: basic data entry, simple aggregations and calculations, and outputs in various electronic formats. Their high-level designs are correspondingly simple and fit for purpose. It is only the large number of data fields per report that makes the applications complex.

However, the design of the database front-end applications could be improved to make them more user-friendly. In this respect, the Microsoft Access-based HMIS application in the D/PHOs is by far the worst, making simple operations like printing a report very cumbersome. The HSIS pilot and central HMIS applications are generally a lot better; a few simple changes would make things a lot easier for users. A combination of user interface prototyping and user testing during the software development process would improve the user-friendliness of future applications.

Relevant Recommendations

- 5.1 STRENGTHENING THE CURRENT HSIS P
- 5.2 REVISED HMIS TOOLS AND DATABASE
- **Error! Reference source not found. Error! Reference source not found.**

3.1.2 Flexibility

The applications have been criticised for being inflexible, as it is difficult to modify the tools and indicators, despite requirements changing over time. This difficulty arises primarily from a combination of the lengthy end-to-end reporting process and ineffective software contractor management. Implementing a new indicator can require the modification and re-distribution of a whole series of software components at the district and central levels. The paper-based tools used at the health facilities can also be impacted.

It is possible to build software applications which are more flexible, allowing new data fields to be added to the user interface and database without the need for software changes. However, this approach is not recommended, as such applications require a more complex internal database structure to enable that flexibility. As a result, report generation, data analysis and integration with other systems become more complex.

Instead, the end-to-end process should be simplified by moving towards a centralised online system into which facility-level reporting data would be entered directly from the D/PHOs. This simplification reduces the number of IT components and systems involved in getting data from facility to the centre, thereby minimising the number of software components that have to be modified to implement a change to HMIS tools and indicators. Such a centralised online system was not possible in the past as reliable Internet connections were not available in most D/PHOs. With the ongoing rollout of Asymmetric Digital Subscriber Line (ADSL) Internet services across the country, the situation has improved dramatically and will only continue to get better.

Relevant Recommendations

- 5.1 STRENGTHENING THE CURRENT HSIS P
- 5.2 REVISED HMIS TOOLS AND DATABASE

- **Error! Reference source not found. Error! Reference source not found.**

3.1.3 Software Bugs

Across all the applications there are numerous bugs that undermine data integrity and require labour-intensive workarounds. In some cases whole areas of functionality were simply never completed by the software contractors. While many bugs could be resolved by small software changes, some bugs were reported years ago but still remain unfixed.

There are two main reasons for this situation:

- Inadequate testing before acceptance, deployment and payment.
- Inadequate ongoing engagement with the contractors.

Relevant Recommendations

- 5.3 HMIS DISTRICT DATA EXPORT QUICK F
- **Error! Reference source not found. Error! Reference source not found.**

3.2 HMIS AND HSIS PILOT TOOLS AND INDICATORS

A consistent set of health service indicators is required from all districts for national annual reporting purposes. However, the HMIS reporting tools differ significantly between HSIS pilot and non-pilot districts. As the health service indicators presented in the annual report come from the central HMIS, data from the monthly reports of the HSIS pilot districts are manually re-entered into that system. However, because of the differing tools, many indicators remain blank for the pilot districts.

Relevant Recommendations

5.2 REVISED HMIS TOOLS AND DATABASE

3.3 HSIS PILOT VS. HSIS VISION

The HSIS pilot is a valid first step towards achieving the goals of the original HSIS vision, given the current environment in which it has to operate. However, it has some severe limitations.

The main limiting factor for the pilot is the fact that most of the systems from which it should ideally be drawing and integrating reporting data are either not fully functional or not easily accessible. As a result, the only way to achieve any sort of integration is to do so manually: measures that should be derived directly from the data in those systems are instead compiled manually and entered separately into the HSIS pilot application.

As a result of this current approach:

- The possible volume and detail of the integrated reporting data are limited by the manual effort required in their compilation.
- The types of reporting and analysis that are possible using these data are correspondingly limited.
- There is a lot of scope for human error during the compilation of the integrated reporting data. This could lead to inconsistencies between HSIS reports and the relevant domain-specific MIS.

Section **Error! Reference source not found.** proposes a phased approach to achieve the ultimate goals of HSIS, starting with the current pilot system and building upon it incrementally.

3.4 IT ENVIRONMENT AND OPERATIONS

3.4.1 Central Server Environment

The central HMIS servers are not located in a dedicated machine room. They are in a normal office, which is air-conditioned but dusty. Access to this office is not very restricted, opening up possibilities for deliberate or accidental interference with the server hardware by unauthorised personnel.

There is a power backup facility, but without enough capacity to guarantee continuity of service during a power outage lasting several hours. Fortunately, the offices are on a power line that is prioritised during periods of generalised power shedding by the electricity supplier. Long power outages are apparently infrequent.

Relevant Recommendations

- **Error! Reference source not found.** **Error! Reference source not found.**

3.4.2 D/PHO Server Environment

There were no dedicated server hardware or machine room environments at the D/PHOs visited during this assessment. The HSIS pilot and HSIS applications and databases are installed on user machines.

Rather than proposing a large investment to create a robust server environment at each D/PHO, the recommendations in this report are aimed more at:

- Moving towards an architecture within which the D/PHO-hosted systems (e.g. the District Health Information Bank (DHIB)) are not critical to the continuity of the end-to-end reporting processes.
- Simplifying D/PHO technical support and operations so that locally hosted systems can be brought back online quickly in the event of a failure.

Relevant Recommendations

- **Error! Reference source not found. Error! Reference source not found.**
- 5.1.3 Modified High-Level Architecture

3.4.3 Networks

Network configurations at the D/PHOs and the central DOHS offices were not examined in great detail during this assessment. However, some possible issues were noted:

- The central servers are not on an isolated and protected Local Area Network (LAN) segment;
- The central servers do not have a dedicated Internet connection and do not have any priority in accessing the connection shared with the rest of the office.
- There do not seem to be any measures in place to ensure appropriate network usage. Users are free to engage in bandwidth-consuming activities such as streaming video and peer-to-peer file sharing.

As things stand, there is no way to guarantee the performance of online access to the central HMIS system or the planned HSIS reporting portal.

Relevant Recommendations

- 5.8 DOHS

3.4.4 Antivirus Measures

Despite reports of problems caused by computer viruses, all computers inspected during this assessment, in the central DOHS offices, D/PHOs, and health facilities visited, had Kaspersky Antivirus software installed with up-to-date virus definitions. However, as the antivirus software is not centrally managed, there is no guarantee that this is the case across the board. Licences may expire, new machines may be installed without antivirus software, and some installations may not be updated regularly.

The Kaspersky product line includes administration tools that provide a central view of all machines on a network: tracking licences, monitoring virus activity, reporting security breaches, and facilitating software and virus definition updates. These tools should be used to ensure effective antivirus protection across the organisation.

In addition, as viruses usually attack vulnerability in an application or operating system, it is important that all systems on the network are regularly updated with the latest available security patches, ideally automatically using services like Windows Update[2].

3.4.5 Backups

Database Backups

The central HMIS application provides a facility to back up and restore the database. However, this does not work. The backups generated are incomplete and if actually used to restore the database would result in data being lost. Fortunately, database backups can be, and are being, taken manually, using the more complex Microsoft SQL Server Management Studio tool.

In general, across all applications, database backups are being taken irregularly and on an ad-hoc basis.

A simple database backup and restore utility should be created, built on SQL Server's native backup functionality. It should be possible for this utility to be executed both manually and through Windows Task Scheduler.

System Backups

System backups encompass not just the database content but the application software and operating system as well. They are used to restore a system on the same or another machine in case of events such as software corruption or server hardware failure.

There are no system backup tools or processes currently in place. In the case of a general system failure, an application server would have to be rebuilt from scratch. The custom developed applications would pose the biggest problem, as they would have to be installed by the original software developers, assuming they still have a copy of the appropriate version of the application. Such failures have already occurred several times, leading to system outages of up to several weeks at a time.

Relevant Recommendations

- **Error! Reference source not found. Error! Reference source not found.**

[2] The possibility of rogue machines coming onto the network is a risk that is very difficult to eliminate in a large office with staff and visitors coming, going and accessing the network over Wi-Fi. However, servers located on an isolated LAN segment, as recommended in section 5.8, and with up-to-date antivirus software and operating system patches, will be well protected.

4 HSIS EVOLUTION

This section proposes a scenario for how HSIS might evolve over time.

4.1 INTEGRATION APPROACHES

The integration of health sector information is a key component of HSIS. There are several possible interpretations of how this might be achieved, each with its own level of complexity and capabilities. The approaches presented below are not mutually exclusive. All three can be combined in a single overarching system to address a particular set of requirements and practical constraints.

Manual Integration

This is the approach that has been adopted by the HSIS pilot districts. Data relating to multiple programmes are manually compiled from paper-based recording tools and captured on the HMIS reporting tool at each health facility. In parallel, some of the data are compiled and entered into additional reporting tools for individual programmes.

Key Benefits:

- Manual integration is the only possible approach if the required source data are not reliably accessible in an electronic format.

Limiting Factors:

- Duplication of effort creating an additional burden for health facility staff.
- Many opportunities for human error and inconsistent outputs.
- Limited capacity – without an increase in staff numbers, only so much data can be compiled this way and hence incomplete data are submitted.
- Limited data analysis possibilities – the reported data are already selected and aggregated, providing a single view of the recorded data with a particular purpose in mind (e.g. the annual health sector report).
- Lack of flexibility – if reporting requirements change, the paper-based tools have to be revised, printed and distributed across the whole country.

Automated Integration

With this approach, report generation software accesses the data required from multiple electronic data sources. Typically, these sources would be pre-existing database systems originally developed to support individual departments, divisions or programmes (e.g. LMIS). However, they can include any electronic data that have been loaded into a database (e.g. survey data).

To avoid having to access each data source directly at the time a report is generated, particularly if continuous online access to all required systems cannot be guaranteed, a separate integrated reporting database would be established. Reporting data would periodically be extracted in bulk from the various data sources and integrated, so as to be available when required for reporting purposes.

Prerequisites:

- Revised data recording tools.
- Access to the underlying electronic data sources.

Key Benefits:

- Can potentially handle an unlimited volume of source data.
- More data analysis possibilities.

- Dramatically reduces the scope for human error and inconsistencies to occur.
- As report generation does not require manual labour, a much larger set of reports can be created for different purposes and audiences.
- Data from different sources can be included in a single report.

Limiting Factors:

- Challenging (but not impossible) to correlate data from different sources at a detailed level.
- Database programming skills are required in order to create new types of reports, providing different views of the source data.

Data Warehousing

A data warehouse is a purpose-built database with a specialised structure, used solely for reporting and analysis. Data sets are loaded in bulk from multiple electronic data sources and integrated to allow every possible correlation between them. Multiple types of reporting and analysis software can then be used to slice, dice, aggregate and explore the complete data warehouse.

Prerequisites:

- Access to the underlying electronic data sources.
- Adequately detailed source data – data warehousing would be particularly powerful in the context of HSIS if the HMIS data gathering tools were entered at health facilities.
- A uniform coding scheme.

Key Benefits:

- All of the advantages of automated integration.
- The ultimate in flexibility for data analysis.
- Report creation and analysis is easier.
- Non-IT users can create their own views of the data.

Limiting Factors:

- Very specialised IT skills are required to design the structure of the data warehouse model.
- The automated loading processes can become complex, and need to be run and monitored regularly by IT staff.
- A successful data warehouse can grow very large over time, demanding more and more hardware and storage space.

4.2 PHASED DEVELOPEMENT

This section is intended to provide a long-term vision for HSIS, starting simply, and evolving over time into an increasingly functional and flexible system. Benefits will be achieved at each phase, even if the ultimate goal is not accomplished. Decisions as to when or whether to proceed to the next stage can be reviewed in light of available resources and whether it is deemed necessary, but it is important that all intermediate steps are guided by a long-term vision.

During each of the indicated phases, there may be multiple releases of the system software, refining and expanding the functionality that is possible during that phase (e.g. revised and new reports).

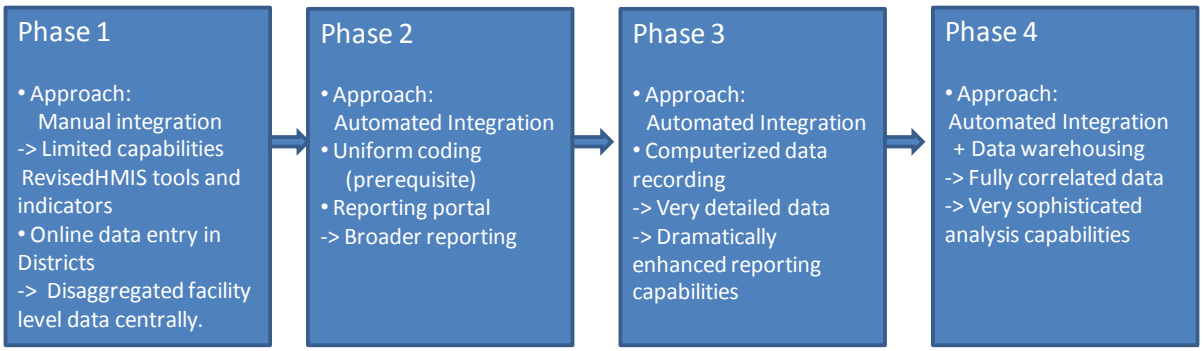

Figure 1 – Phased Development

Phase 1 – HSIS Pilot

Approach: Manual Integration.

- Quick wins:
  The pilot application is fixed to work as originally specified (see Section 0 –
  - Bug Fix Initiative).
  - Data entry at the D/PHO via a central online application in HSIS pilot districts.
- Revised HMIS data recording tools are introduced in HSIS pilot and non-pilot districts.
- A corresponding new HMIS reporting data entry application is developed, replacing the HSIS pilot application.
- Disaggregated facility-level data for HSIS pilot districts are available centrally (once deemed successful this can be scaled up to the remaining 72 districts).
- There is no DHIB yet. For the time being, equivalent functionality is provided to the D/PHOs by the central online application.
- Integration with the central HMIS system to automate the transfer of relevant district-level data for the national annual report.

Phase 2

Approach: Automated Integration.

- A stand-alone reporting portal is introduced, accessible to many stakeholders.
- Uniform coding is introduced, implemented as a central Master Reference Database (MRD) (see Section **Error! Reference source not found.**). This is a prerequisite for adopting the automated integration approach.
- DHIB is introduced for HSIS pilot districts (once deemed successful this can be scaled up to the remaining 72 districts).
- A broader set of reports is now feasible.

Phase 3

Approach: Automated Integration.

HMIS data gathering tools are computerised at the health facility level (see Section 5.4 – COMPUTERISING HMIS DATA RECORDING T). As a result:

- Health facility staff no longer have to compile reports manually and data entry can be completed at facility level.
- Electronic data can be extracted in bulk from the health facility application, sent to the D/PHO, and uploaded into the central HMIS database.
- The source data available via HMIS can be as detailed as desired.
- Reporting and analysis capabilities are enhanced dramatically.

- Reporting for programmes such as Aama and Emergency Obstetric Care (EOC), which use the same ultimate source data as HMIS, could be rolled into HSIS, being delivered through the same reporting portal.

Phase 4

Approach: Automated Integration + Data Warehousing

- The benefits of the investment in Phases 2 and 3 are retained.
- Where adequate Internet connectivity is available, the collection and consolidation of detailed HMIS data from health facilities is automated, network connectivity permitting.
- Data warehousing is introduced, starting with a limited dataset but being extended incrementally over time.
- Reporting and analysis capabilities become as good as they can get.[3]

4.3 PROPOSED TARGET ARCHITECTURE

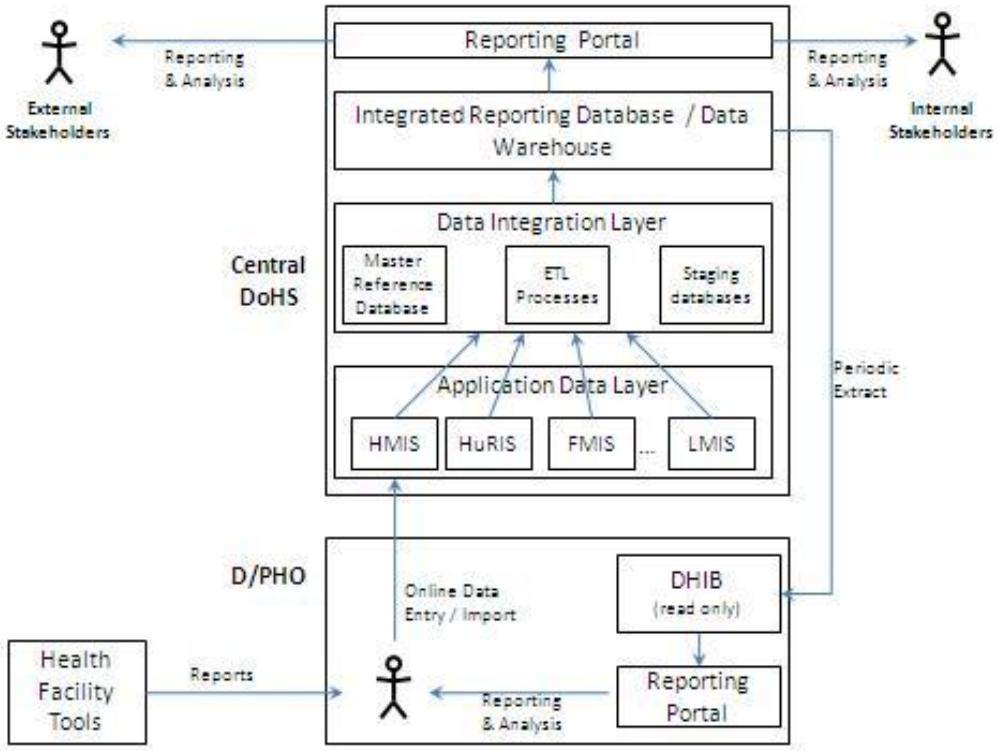

Figure 2 – Target Architecture

Figure 2 above presents the proposed target architecture for HSIS. The key points to note are that:

- Data integration is a central function, performed by a set of Extract, Transform and Load (ETL) processes (see Section 5.6 ETL). Although it would be theoretically possible to have an alternative architecture with data integration occurring between systems at the D/PHO level, the excessive burden of operating and monitoring an identical set of complex components at each of the 75 sites makes such an approach unfeasible in practice.

[3] While it was not directly observed during this consultancy, at present, a lot of higher-level hospitals are reported to enter their own data (using their own individually developed databases, different from the HMIS database) and send it to the MIS section. Where this is the case, for Phases 1 and 2, they should send HMIS 34 reports to the D/PHO. From Phase 3, they should either modify their application to export data in the same format as the standard health facility application, or adopt that standard application.

- The primary function of the MRD is to support a uniform coding system enabling the linking of data from different sources (see Section **Error! Reference source not found. Error! Reference source not found.**).
- The DHIB and its reporting portal are included in the original HSIS strategy. The D/PHO reporting portal is an exact replica of the central one. Similarly, the DHIB is a replica of the central reporting databases, albeit containing the data for just one district to enable local area planning.
- The data in the DHIB are refreshed periodically from the central reporting databases. These data are never modified in the DHIB or fed back to the central databases – such a bi-directional synchronisation scheme would be unmanageable.
- The DHIB refresh process could be implemented using ETL processes or database replication functions combined with a reliable and efficient wide-area replication service such as Dropbox.
- HMIS is just another application alongside the other health sector MISs.
- HMIS data entry is performed at the D/PHOs via a single online HMIS application.
- At first glance some elements of this architecture may seem to clash with the original HSIS strategy document which calls for an "integrated, comprehensive and decentralised health sector information system." However, that document was written at a time when a decentralised function like health service data entry could only be achieved using distributed systems. Subsequent advances in network technology and infrastructure have blurred the distinction between centralised and decentralised IT systems. In this case, decentralised functions are supported by centrally hosted systems for reasons of efficiency and sustainability.

5 RECOMMENDATIONS

5.1 STRENGTHENING THE CURRENT HSIS PILOT

5.1.1 Bug Fix Initiative

An initiative should be launched to fix the known bugs in the HSIS pilot application. A new service contract with the software contractors may be required; if it is, the new contract should have stronger incentives/penalties. It should also avoid the same pitfalls that last time allowed the contractors to be paid before the system was completed and thoroughly tested. A cut-off date should be set for completion of the initiative, and if the contractors do not meet this deadline, the penalty clauses should be applied and the contract terminated so that resources can be fully focused on other more strategic initiatives.

5.1.2 MIS Software Manager

A software management role should be clearly defined and assigned to an appropriate current member of the MIS section in order to ensure proper engagement with software development contractors. The MIS Software Manager's responsibilities would include:

- Acting as liaison between developers and system users and stakeholders.
- Resolving conflicting software change requests.
- Formally signing off contractor deliverables once acceptance criteria are met.
- Analysing and validating reported bugs before they are raised with the contractors.
- Actively tracking contractor progress.
- Coordinating software acceptance testing.
- Software change management – ensuring that the number of co-existing software versions is kept to an absolute minimum and that they are merged over time into a single version.
- Release management – minimising the frequency of upgrades to the live systems by batching changes together.

5.1.3 Modified High-Level Architecture

Once fixed, the HSIS pilot application should be deployed centrally. This would then be accessed online from the D/PHOs for reporting and analysis purposes. Such online access of a central system from the districts has already been successfully achieved for LMIS.

As a result of this change, the current DHIBs in the districts would be shelved temporarily. However:

- The exact same functionality provided by a DHIB would be provided to the districts by the online application (i.e. basic facility-level data reporting and analysis).
- Disaggregated facility-level data would now be available at the central level.
- These data could be made accessible to a broader audience.

Some technical activities will have to be undertaken to make this a success:

- The data already in the three existing district databases should be merged into a single central one. Microsoft SQL Server has tools to facilitate this process (SQL Server Integration Services).

- A small application and database change would be required to ensure that users can only enter and modify data for their own district.
- The new architecture should be performance tested to determine whether the bandwidth of the central server's Internet connection needs to be increased. This bandwidth should be dedicated to that server and it may be necessary to ask the Internet Service Provider (ISP) to reverse the usual upstream/downstream bandwidth balance.

5.2 REVISED HMIS TOOLS AND DATABASE APPLICATION

5.2.1 HMIS Tools

An initiative should be launched to review and revise the HMIS tools to ensure that they meet current health service reporting needs. These tools should be deployed to all districts, (HSIS pilot and non-pilot), for the start of the next reporting year (July 2012).

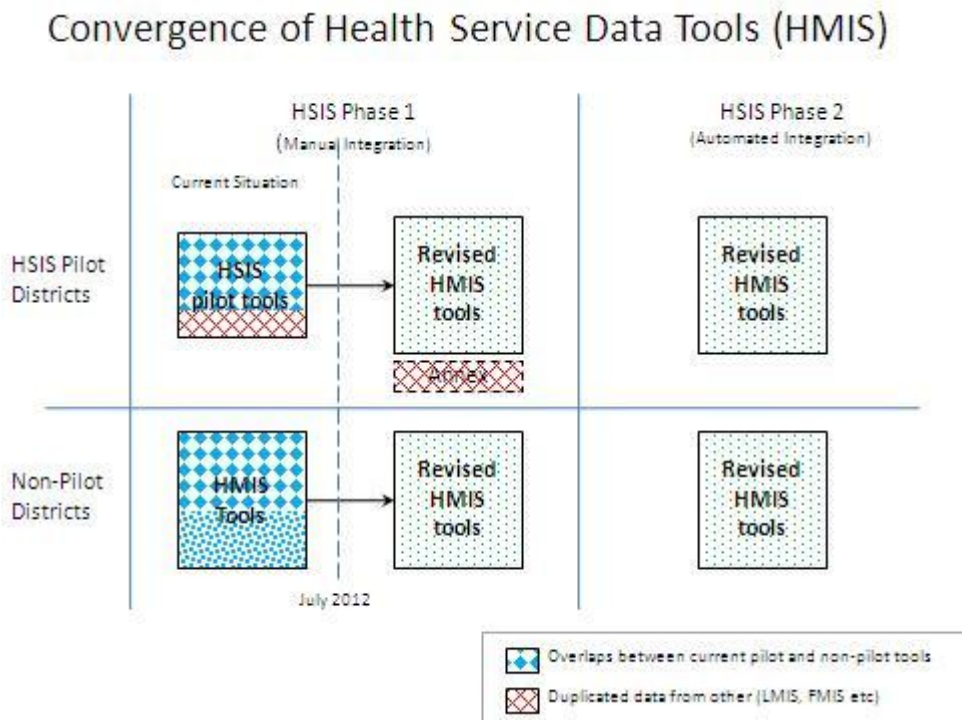

Figure 3 – Convergence of Health Service Data Tools (HMIS)

In HSIS pilot districts a small set of additional questions/tools may be used to capture facility-level reporting data that would normally be the responsibility other systems (e.g. FMIS, HuRIS) which are not yet in a state allowing the overall HSIS reporting system to integrate with them directly. Once such automated integration becomes possible (HSIS Phase 2) this additional "Annex" of reporting tools/questions would become redundant.

5.2.2 HMIS Database Application

A corresponding new HMIS database application should be developed in time for the start of the next reporting year. It should be a single application which can be deployed both centrally and in the districts, HSIS pilot and non-pilot. Online data entry to the central system should be piloted in HSIS pilot districts

first; non-pilot districts should use a local deployment of the system until online data entry has been successfully piloted. To enable these two different types of deployment certain system features are required:

- Configurable *roles* can be assigned to individual user logins, controlling which application screens they can access and what actions they can perform (e.g. access to specific data entry, reporting and administration screens). A single user may be assigned multiple roles.

- Configurable *permissions* can be assigned to individual user logins, controlling the scope of their access to data, what they can view and what they can update (e.g. specific districts, all districts).

- Automatic generation of district-level aggregations when the underlying facility-level data are available. This process would be triggered and possibly scheduled by a system administrator. It would scan the database for all districts and create or update aggregated data records as appropriate, using the latest facility-level data.

- Export and import functionality for aggregated district-level data, much like the current system.

- Separate, easily removed database tables and data entry screens for HSIS Annex, access to which is controlled by the previously mentioned roles and permissions.

- Measures to facilitate online data entry over the Internet, for example:

    - Automatically saving data entered by users as they move from tab to tab in a large data entry screen so that their work is not lost if the Internet connection is disrupted before data entry is complete.

    - Application tuning to ensure that it is as responsive as possible over slower Internet connections (e.g. avoidance of bandwidth-consuming graphics, use of AJAX for screen updates).

With this system, the only technical distinctions between HSIS pilot and non-pilot districts would be that:

- Users in pilot districts have the roles and permissions required to use the HSIS Annex data entry screens.

- Users in the pilot districts would access the online central system for data entry. Note that for reporting and analysis purposes, however, they can access a local read-only DHIB.

5.3 HMIS DISTRICT DATA EXPORT QUICK FIX

It is not worth expending too much effort fixing all of the problems of the current HMIS application in the D/PHOs if it is soon to be replaced as per the previous recommendation. However, one particular software bug, probably requiring just a small modification to fix, is worth singling out for immediate attention.

The application has a facility whereby district reports (HMIS-33) are exported in an Extensible Mark-up Language (XML) format. These can then be sent by email to be imported into the central system. This export-transfer-import process worked well with the original SQL Server application in the districts. However, this process has been broken ever since the original application was replaced by the current one. As a result HMIS-33 reports received from the districts now have to be re-entered manually into the central system.

The reason this problem has not been fixed to date may be because it requires the collaboration of two different software contractors: the developers of the central HMIS system developers to troubleshoot and identify exactly what is wrong with the XML files being imported, and the district application developers to correct the export facility accordingly. This collaboration could be coordinated by the MIS Software Manager (See Section 5.1.2 MIS Software Manager).

Fixing this problem would relieve the central MIS staff of a heavy data entry burden and free them to pursue more productive activities.

5.4 COMPUTERISING HMIS DATA RECORDING TOOLS

It should be a long-term goal to computerise the HMIS data recording tools used at health facilities. This would relieve health facility staff of the current burden of producing the monthly HMIS reports. In addition, it would make a whole new level of electronic data available for computerised reporting and analysis, enabling the third proposed phase HSIS evolution. It would also reduce human error in the current aggregation at each level, and, as mentioned above, reduce the burden at the central level.

This would obviously be a major project with the rollout to all facilities being a gradual process, but it may not be as daunting as it might initially sound:

- All hospitals and about 25 of the primary health care facilities in the country already have computers and Internet connections.
- The computerisation could start small with something like an Excel version of the master register.
- A free open source product like Open MRS[4] could be used to build a more complete health facility application. Open source products might also require additional software or database management systems (e.g. Apache or My SQL) and therefore additional expertise to implement/maintain. However, regardless of whether a system is custom developed or implemented using an existing product, resources and expertise will be required, probably through external contractors. One advantage of open source products over custom development is that one is not tied to a single contractor (i.e. the original developer) for subsequent updates or changes.

5.5 OPEN SOURCE SOFTWARE

Before undertaking the custom development of any new software applications, the open source market should be explored to see if a free product is available that already supports the bulk of the required functionality. An open source product may be more complex than a custom designed application and may require some additional customisation to meet requirements fully, but it would provide a tried and tested framework that can be built upon to create a robust and flexible system.

Throughout this report, various candidate open source products are suggested where appropriate.

5.6 ETL SOFTWARE TOOLS

What does ETL mean?

Achieving the vision of HSIS hinges upon having the ability to access, move and integrate a large volume of data from many systems. These activities are performed by what are collectively called ETL processes, implemented using software components that extract data from a data source, transform it to integrate

[4] One of the evaluation criteria when selecting an appropriate product would be to consider how the data integration components of the architecture can extract reporting data and how they are structured.

with data from outer data sources, and load them into a dedicated database from where they can be accessed by reporting and analysis tools. These processes have to be run regularly and monitored closely to ensure the timeliness, completeness and integrity of data available for reporting.

How should ETL components be implemented?

The current HSIS pilot and HMIS applications have some rudimentary ETL processes built into them, in the guise of the report export and import modules. These software modules have been custom developed. This approach has severe limitations:

- Flexibility – To add a new ETL process or modify an existing one, the application that is the source of the data has to be modified and re-deployed. Aside from the software development effort, even the smallest change requires the collaboration of at least two groups of software developers, those of system(s) exporting the data and those of the system receiving it.
- Scalability – With the current export and import components, non-technical application users are initiating and controlling the implicit ETL process, record by record. This will quickly become unmanageable and unreliable as the volume and detail of data to be integrated increase.

ETL processes should be implemented as stand-alone components using an ETL software product specifically designed for this purpose. Such products require database programming and data modelling skills to create ETL processes but the amount of programming actually required is minimal or frequently even completely eliminated. They provide a toolbox of pre-existing components that can be strung together to create almost any conceivable ETL process. They allow data to be manipulated in bulk and provide powerful error handling facilities. The resulting processes can be completely automated and scheduled to run regularly.

What specific ETL products are appropriate for HSIS?

The standard and Enterprise editions of Microsoft SQL Server have a built-in ETL tool, called *SQL Server Integration Services (SSIS).* This could be a candidate for the HSIS programme as the current HSIS pilot and HMIS systems are built using Microsoft database products (MS Access and other database support).

Pentaho Data Integration is a free open source product that is even more flexible than Microsoft's SSIS, supporting all popular database products (Microsoft, Oracle, MySQL, Postgres etc.) and operating systems (Windows, Linux etc.).

Talend Open Studio is another excellent free data integration product.

Other proprietary products are available but these tend to be rather expensive (e.g. Informatica).

The use of ETL products does not eliminate the need for a software development contractor. However, they reduce the development effort as the contractor can use them to build ETL processes quickly that would take a lot longer to develop from scratch. An additional advantage of using open source ETL products is that the contractor can provide ongoing support with the open source community as a backup.

5.7 HOSTING

The continuous availability of the central HMIS and HSIS components will be critical to the success of the overall HSIS programme. These should be hosted in a data centre environment providing a range of common facilities and services:

- Environmental control: temperature, humidity, air quality.

- Physical and network security.

- Backup power: Uninterruptible Power Supply (UPS) and generator.
- Hardware redundancy and failover mechanisms.
- Disaster recovery site and processes to provide continuity in the case of fire, flooding or other Acts of God.
- Backup and restore facilities.
- Server provisioning and resource allocation.
- Server virtualisation.
- 24/7 service monitoring and support.

Given the huge gap between the current situation and this ideal, and the great cost of such a data centre, it is probably appropriate to adopt an incremental approach to improving hosting facilities.

1. ***Third-party commercial hosting (optional)*** – International companies such as Rackspace offer excellent hosting services providing all the desired features listed above for a fraction of the cost of setting up a comparable private data centre. This is a viable long-term alternative if government policies do not preclude the storage of government data on externally hosted systems. The central HMIS system is currently hosted in this way, via the local software contractor – the system is being provided more as a service than as a software package.

2. ***MIS server room*** – A small server room could be built within the MIS section, limiting physical access to the hardware and providing an air-conditioned and dust-free environment. Basic server virtualisation, system backup tools and redundant server hardware should be used to enhance server availability. However, if MIS were to move premises, as is specified in the HSIS Strategy, resources may be wasted on this set-up.

3. ***Consolidation of department IT resources*** – There are several systems scattered across the DOHS, tucked away in different offices. The LMD has just created a small server room with air-conditioning, racks and a dedicated Internet connection to host their new eBidding system and LMIS. It would be to the mutual benefit of all divisions to cooperate, sharing or pooling their IT resources to create a more robust hosting infrastructure. Divisions with spare capacity in their server rooms could host the servers of less well-equipped ones, in return for cost sharing. Divisions with separate server rooms could provide failover facilities for each other.

4. ***NHIC-managed data centre*** – The process of consolidation could eventually lead to the de facto creation of a department- or ministry-wide data centre. This would be very much in line with the creation of the National Health Information Centre (NHIC), as described in the HSIS national strategy document. There is no specific requirement for the NHIC to have its own dedicated data centre. The Department of Science and Technology has just completed building the Government Information Data Centre (GIDC) which is intended to be a shared resource available to all government departments. Although not yet fully operational, it should be considered as a long-term hosting option, or at the very least as a disaster recovery site.

5.8 DOHS NETWORK

If the central HMIS and HSIS servers are to continue being hosted in the MIS section, some modifications should be made to the DOHS network. Figure 1 below illustrates the recommended changes.

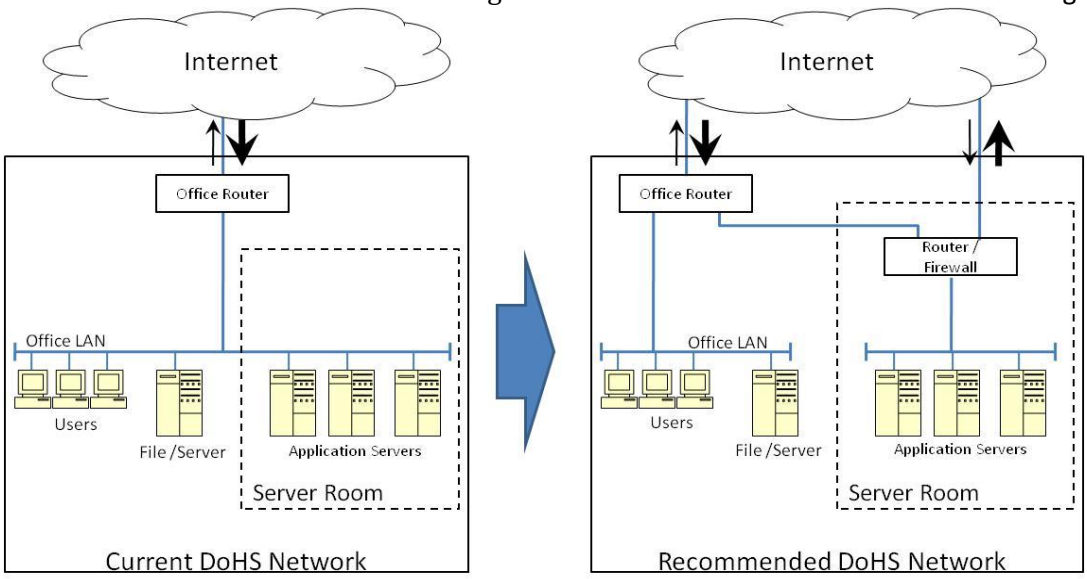

Figure 4 – Proposed Changes to the DOHS Network

- Application servers should be on a LAN segment isolated from the rest of the office by a router with good firewall capabilities.
- The server room router should have two Wide Area Network (WAN) ports, one to connect to the Internet, and the other to connect to the rest of the DOHS internal network. The dedicated WAN port would have a public IP address. The other would have an IP address on the office network. Static routing rules would direct traffic over the appropriate WAN interface, and implement failover.
- The server room should have a dedicated Internet connection.
- On most Internet connections, traffic to the Internet (upstream) is much less than traffic from it (downstream). This is because web page requests are small while the responses, full web pages, are much larger. ISPs usually take advantage of this by allocating less bandwidth for upstream traffic – A 512kbps connection might actually be 512kbps downstream but only 64kbps upstream. This balance should be reversed for the server room Internet connection.[5]
- The server room router should be configured to:
    - Close all Transmission Control Protocol/Internet Protocol (TCP/IP) ports except those required to access the functionality provided by the application servers (e.g. ports 80 and 8080 for access to web applications).
    - Have a public Internet Protocol (IP) address allocated by the ISP.

[5]Astandard ADSL configuration would not provide optimal performance for this configuration; collaboration with the ISP would be required to address this.

- Use Network Address Translation (NAT) functionality to forward requests from the Internet to the appropriate internal application server and port.

- Use the main office Internet connection as a backup in case the dedicated Internet connection fails.

5.9 VIRTUALISATION

What is virtualisation?

Virtualisation is a method of isolating operating systems and applications from the hardware on which they run, and transplanting them to a virtual machine created and managed by the virtualisation software. Multiple virtual machines can be run independently and at the same time on a single physical machine. An individual virtual machine can be halted and resumed later, picking up exactly where it left off. It can be moved to another physical machine without applications having to be re-installed or even restarted. A complete virtual machine can be backed up easily and restored to its original state at a later time.

What would the benefits of virtualisation be for HSIS and MIS in general?

- *Availability* – With the ease of backing up, restoring and moving complete virtual machines, outages due to system corruption or hardware failure could be kept to minutes rather than weeks, as is currently the case.

- *Efficiency and scalability* – Multiple virtual servers could initially be run on a single physical server, without fear of them interfering with each other, maximising the utilisation of the current hardware. If demands for processing power and memory resources increase beyond the capacity of a single machine as the usage and functionality of applications increase over time, the virtual servers can be easily separated to run on separate new physical servers.

- *Software deployment and support* – The original SQL Server-based HMIS application deployed at D/PHOs had to be abandoned because its components were too difficult to maintain and support. As local IT support capacity was limited, the system developers had to travel to each D/PHO whenever software updates were required. In addition, users had a lot of scope for breaking the HMIS application, e.g. by installing unrelated but conflicting software on their machines or by accidently deleting required files.

  With virtualisation, a dedicated virtual machine could be created and maintained centrally for use at the D/PHOs, with all required software pre-installed and configured. Users would be granted minimal access rights, preventing them from modifying critical components. A single such virtual machine could be used as a template for creating a separate virtual machine for each D/PHO; it would be cloned many times and the configuration of each copy modified slightly to reflect the environment of a particular D/PHO. The customised virtual machine could then be sent to the relevant D/PHO and simply started up.

How can it be done?

Virtualisation can become very complex and expensive. VMware provides the most popular virtualisation tools on the market. This is proprietary licensed software but there are free entry-level versions that can provide all the features mentioned above:

- VMware vSphere Hypervisor (ESXi) – A basic set of easy-to-use tools to convert existing systems to completely virtual machines, to create additional new virtual machines, and to operate all of them in parallel.
- VMware Player – an even simpler alternative for D/PHO installations, allowing the original user's machine to remain un-virtualised while running a previously created virtual machine in parallel.

Note, however, that the user machines may require memory upgrades to optimise the performance of the virtual machine.

The Ubuntu Linux distribution has an in-built and very powerful free virtualisation product called KVM. However, this has not yet been widely adopted because its management tools are not as complete or as easy to use as VMware's. This will change over time and KVM is likely to become a strong competitor in this market.

5.10 UNIFORM CODING SYSTEM

A uniform coding system enables the linking of data from multiple systems for computerised reporting and analysis. The multitude of MISs within MOHP do not share such a system. The future evolution of HSIS requires that one be put in place.

The initial definition of a uniform coding system need not be a complex matter. Ensuring that it is implemented in all existing and new systems is a long-term process. In addition, the coding system has to be maintained carefully to accommodate new requirements over time. The uniform coding system will be managed using a MRD.

Annex 1 provides more technical details on the implementation of a uniform coding system.

5.11 SOFTWARE DEVELOPMENT PROCESS

A software development methodology should be introduced to ensure that all custom developed software is fit for purpose, of a high quality, adequately documented, and maintainable in the future. It is particularly important that the software methodology is integrated into all software development contracts and strictly enforced. Annex 2 provides a sample software methodology.

5.12 REPORTING PORTAL

The reporting portal should not be custom developed from scratch. A free open source product such as the Pentaho Business Intelligence Server should be used as a starting point. Such products provide a ready-built framework and web application front end. They can be used to integrate and control access to reporting and analysis components build using a variety of tools. Such components can include:

- Dashboards.
- Online Analytical Processing (OLAP) views.
- Report templates built using report design tools.
- Ad-hoc reporting tools.
- Custom developed web pages.

The Pentaho BI suite of products includes tools facilitating the development of all of the above. It also has facilities to automate the scheduled execution and distribution of reports by email.

5.13 DISEASE SURVEILLANCE AND SENTINEL REPORTING

For disease surveillance, health facilities provide regular updates on the numbers of cases for each of a set of pre-determined communicable diseases. It is crucial that these data are gathered, consolidated and reported on rapidly.

Sentinel reporting is similar except that more detailed data are gathered from a representative sample of hospitals. These data are analysed to identify trends and to provide a basis for setting health sector priorities.

Data from these sites should be entered online into a central database, which, in terms of the HSIS architecture, will reside in the Application Data Layer. The reporting portal can then be used to provide immediate countrywide access to custom developed analysis reports and dashboards.

Where a small volume of data is gathered frequently from each health facility (e.g. less than 20 measures per submission), structured SMS messages sent by mobile phones provide a quick and easy method to get these data directly into the central reporting database. With a free open source SMS gateway such as Frontline SMS or Rapid SMS such a data collection system can be set up in days.

5.14 HUMAN RESOURCES

The table below indicates the types of roles and skills that will be required to develop, operate, and support the systems implementing the HSIS strategy. The specific role names and responsibility allocations are indicative and could be allocated differently.

Role	Responsibilities	Required Skills / Training
D/PHO		
IT Officer (to be covered by existing computer operator in place at each D/PHO)	• First line technical support • System and data backups • System installation • DHIB refreshes • Data entry/import	• Application usage (data entry and reporting portal) • Application administration • Application installation • Backup and recovery procedures • Technical support procedures
Statistical Officer (currently in place)	• Data entry/import • Data validation • Data analysis	• Application usage (data entry and reporting portal) • Validation and feedback processes • Analysis skills and tools
Central		
MIS Software Manager (see Section 5.1.2) (to be covered by the current MIS IT Officer; see Section 5.1.2)	• Software contractor management • Requirement gathering • Functional specifications • Test management • Release management • Change management	• Software development methodology • Source control tool • Test management tools • Issue tracking tools
MIS System Architect (proposed UNICEF national consultant)	• Technical specifications • Review technical designs • Data integration and data warehouse design • Infrastructure design and project management	• Data modelling • Data warehouse design • ETL design tools • Networking • Virtualisation • Data centre technologies
MIS Support Staff (to be	• Second line technical support for	• ETL execution and monitoring

covered by existing MIS section staff, although would be workload-dependent)	D/PHOs • Central systems administration	tools • Database management tools • Application usage (data entry and reporting portal) • Application administration • Application installation • Backup and recovery procedures • Technical support procedures • Helpdesk tools Virtualisation tools
MIS Statistical Officer	• Data entry/import • Data validation • Data analysis	• Application usage (data entry and reporting portal) • Validation and feedback processes • Analysis skills and tools

6 HIGH-LEVEL PLAN

The following Gantt chart presents a proposed high-level plan, based on the approach and recommendations detailed in this report.

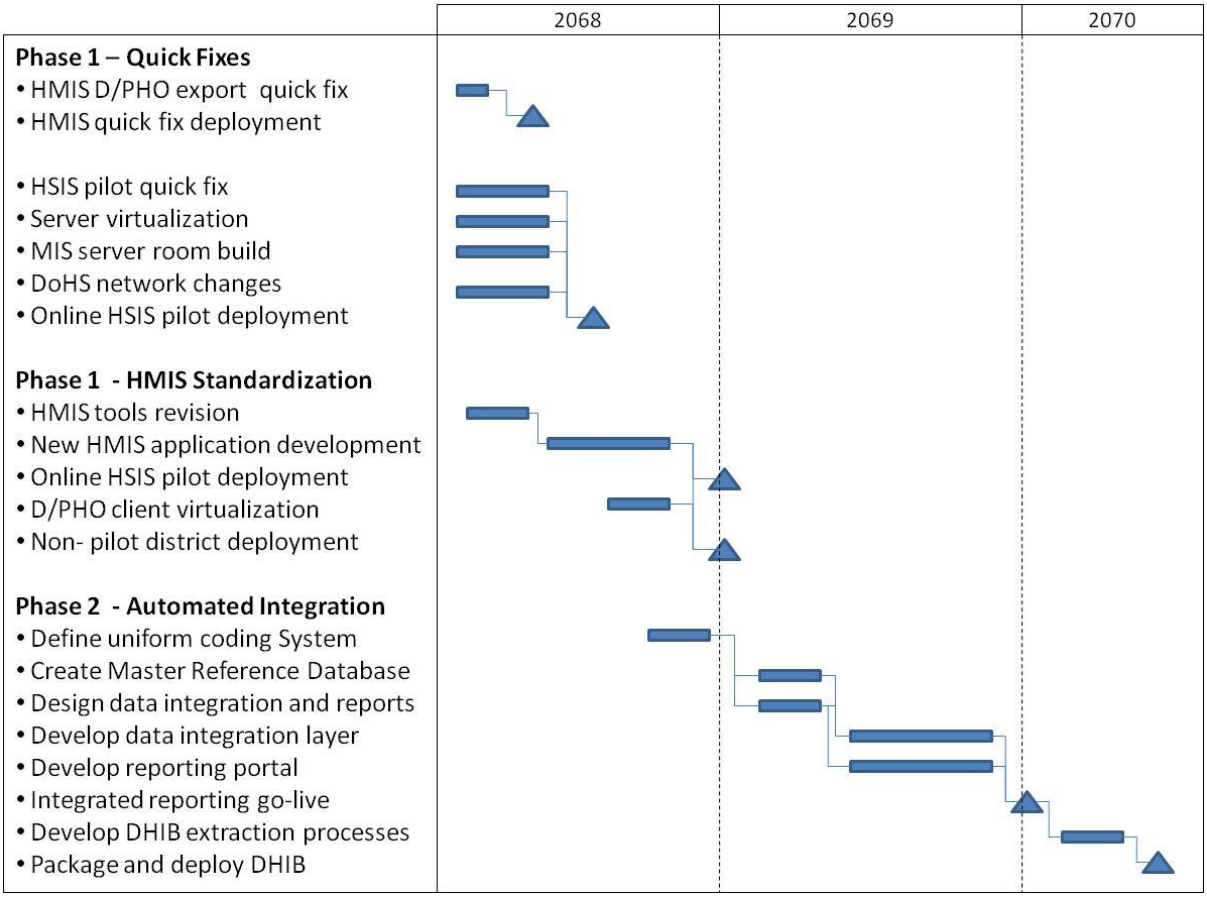

Figure 5– High-Level Plan

ANNEX 1 – IMPLEMENTING A UNIFORM CODING SYSTEM

What should the uniform coding system cover?

In terms of database systems the coding system would apply to two types of data:

- Master Data – lists of persistent entities about which the system was designed to store information, e.g. health facilities.
- Reference Data – lists that are used to categorise the information in the database, e.g. health facilities may be categorised by *facility type*; numbers of cases may be categorised by *disease.*

Not all master and reference data in the systems need uniform codes, only those that will be used to link data from different systems.

How should the codes be defined?

An individual code can be completely arbitrary, as long as:

- It refers uniquely to one thing.
- Once defined it will never be changed.
- It does not have semantics implying relationships or properties which may change over time. For example, for a particular sub health post in Dang the code "DngSHP006" would be inappropriate because if were subsequently upgraded become a health post the "SHP" in the unchanging code would be misleading.

How can the uniform coding system be implemented and maintained?

A simple MRD should be created, containing one table for each set of codes. Each table would have at least two columns: one for the code itself and the other for the name of the thing to which it refers. Tables for codes referring to master data may have additional columns for linking to reference data and for tracking changes to those links.

For example, the table for health facility codes might look like this:

Code	Name	District	Type	Valid from	Valid to	Current
HF0001	Patan	D001	HFT01	01/03/2007		Yes
HF0002	Mental	D001	HFT01	01/03/2010	01/03/2011	No
HF0002	Mental	D001	HFT02	01/03/2011		Yes
HF0003	Dailekh	D023	HFT02	01/03/2008		No
...	...	...	...	...	...	...

Maintaining such a record of changes to master data records is required so that accurate historical reports can be generated even after the attributes of the object referred to by a universal code have changed (e.g. promotion of a sub health post to a health post).

The maintenance of the uniform coding system and the supporting MRD should be the responsibility of a single central authority. Procedures, manual or automated, should be put in place to ensure that changes in the MRD are propagated to all relevant systems.

How can the uniform coding system be adopted by all the relevant systems?

The long-term goal should be for all relevant systems to be modified to use the uniform coding system. This entails modifying master data records in each system's database to include the uniform codes alongside the system-specific codes. The system-specific codes can be retained and the application front end and reports need not be changed, unless there is an explicit need to make the universal codes visible to the user.

It may be that such changes to an individual system are not implemented until quite some time after the introduction of the uniform coding system. To avoid a specific new software release, the system owners may wish to wait until other functional changes are necessary. During this interim period, it is still possible to achieve data integration using the uniform coding scheme. The MRD would temporarily contain another set of tables, one for each system to be integrated, mapping the new uniform codes to the existing system-specific ones. ETL tools can then be used to extract the required reporting data from each system, use the MRD mapping table to change the system-specific codes to the uniform ones, and then load these transformed data into an integrated reporting database.

ANNEX 2 – SAMPLE SOFTWARE DEVELOPMENT METHODOLOGY

A sample methodology is outlined in the following sections, couched in contractual terms:

Specification

1. The client will provide the contractor with a written specification outlining the functionality to be developed. This document will detail the system requirements from a user perspective. It will also specify non-functional requirements such as system capacity, performance and availability targets.
2. The specification may also contain notes dictating or suggesting technical solutions and referring to previously developed components, which it may be appropriate to copy, adapt and reuse.

Analysis

1. The contractor will analyse the client's requirements, as detailed in the specification document.
2. The contractor will then produce a high-level design document, the purpose of which is to build on the specification and to confirm the proposed functionality of the system.
3. The high-level design will present the proposed functionality in terms of typical usage scenarios and screen mock-ups.
4. The high-level design will also include a proposed work plan, detailing milestone dates, and the estimated effort, in terms of consultant days, for the work package to be completed.
5. A user interface prototype, not requiring any real back-end functionality, will be created for the purpose of eliciting early feedback on user-friendliness from real users.
6. The client will review the high-level design and, once satisfied with the content, sign it off, thereby agreeing to the proposed functionality, work plan and effort estimates.

Technical Design

1. Once the client has signed off the high-level design, the contractor will proceed into the technical design phase, the output of which will be a technical design document.
2. It may be appropriate for the contractor to conduct some technical proof-of-concept prototyping at this stage.
3. The technical design will detail the structure of the proposed system, identifying the new software and database components to be created, providing an overview of how each will function and how they will interact.
4. The level of detail for the technical design will be such that the names, inputs, outputs and externally visible behaviour of all software objects are specified. In addition, the physical data model will be documented, detailing all database tables, their columns and relations between them. Internal programming details (i.e. pseudo code) need not be included in the technical design – these should be covered by comments in the code as it is developed during the implementation phase.

Implementation

1. Once the client has signed-off the technical design, the contractor will proceed into the implementation phase, creating the components detailed in the technical design. The output of the build phase will be unit tested.
2. The purpose of unit testing is to ensure that each individual component developed functions as specified in the technical design, and that it appropriately handles all possible errors in input data.
3. During unit testing, the contractor will test each component with typical, maximum, minimum, out of range, and invalid (e.g. wrong data type) values for each data input.
4. Upon completion of implementation and unit testing, the contractor will load all source code, including scripts for the creation of database objects, into a source control system.

5. From this point onwards, the source control system will be used to track all changes made to the source code.

Formal Test

1. The purpose of the formal test phase is to verify that the end-to-end functionality of the developed system complies with the high-level design.
2. The contractor will develop a set of formal test scripts, to be reviewed and approved by the client. These test scripts will detail user actions for end-to-end scenarios, data to be input, and expected system behaviours and outputs. They must explicitly validate all the functional and non-functional requirements identified in the specification and analysis phases.
3. At the beginning of each formal test cycle, the contractor will install the latest version of the developed components (taken from the source control system) into a test environment.
4. As the formal test scripts are executed by the contractor, all problems identified will be documented.
5. Once all the formal test scripts have been completed, or executed as completely as possible, the contractor will implement and unit test fixes for all the problems identified.
6. The contractor will then check all unit tested fixes into the source control system, allocating a new build number to the latest versions of all the components in the source control system.
7. Another formal test cycle will then be executed, using the latest build source code from the source control system (back to step 2).

6.1.1 The formal test phase will end once a single build from the source control system has successfully passed all the formal test scripts.

Delivery

1. Upon completion of the formal test phase, the contractor will deliver a package to the client, containing:
    a. A release of the system built from the latest source code from the source control system.
    b. Release notes providing instructions for the installation of the build.

Acceptance Test

1. The client will install the release package in their own test environment and execute their own validation tests, involving real users as appropriate.
2. In addition, the client will review the received source code.
3. The client will inform the contractor of any problems identified during the acceptance test.
4. The contractor will fix and unit test these problems, include the fixes in a new build in the source control system, perform regression testing (using the formal test scripts) and then deliver another release package to the client.

Final Sign-off

Once a single release package has passed the client's acceptance tests, the client will give final sign-off, thereby confirming the successful completion of the work package.