Goal of the current document

This document aims to give a high level description of the GEMS data exchange module to be built in spring 2011. This information will give partners and clients an idea about the possibilities of data exchange with GEMS.

GEMS data exchange

GEMS is a web application for storing and tracking medical measurement data. GEMS is short for Gems Tracker or Generic Medical Survey Tracker. The tracker "knows" what measurements should be stored when for which patient.

Data in GEMS is stored in two databases: one database containing private patient information and one or more databases containing medical measurement information. The two are connected through internal patient keys not known to users of the application.

The user interface of the GEMS web application concerns itself only with entering patient data and assigning and tracking measurements for those patients. Existing web applications can be used for the manual entry of measurements, e.g. Lime Survey - http://www.limesurvey.org/, or the entry forms can be programmed using the tools provided GEMS.

Possibilities for data exchange include – but are not limited to:

- Importing bulk patient data,
- Importing patient data on demand,
- Importing measurement data,
- Exporting measurement data.

On patient data

A standard GEMS installation commonly needs patient data that is already entered in an Electronic Patient Dossier (EPD). To avoid having to duplicate this information in GEMS the patient data could be imported in its entirety from the EPD and updated regularly afterwards. Usually this is not sufficient as a) patients may be entered between update intervals and b) the amount of data to import may be too big or c) security concerns forbid the indiscriminate importing of all patient data.

A common GEMS installation must therefore import specific patient data on demand and must check for updates on those patients on a regular basis. To import the patient data on demand the GEMS server requests the data from the source after a unique patient identifier (i.e. social security or patient number) was entered. The check for updates can be initiated by either the GEMS server or the source server.

On measurement data

Measurement data is usually exchanged either manually or on a regular schedule. GEMS can either be the source of the data or import the data e.g. from measurement systems.

Measurement data can also be exported on request, e.g. when measurement data on a patient is looked up in and EPD.
Technically it is also possible to import data automatically when requesting data. As the manual entry of measurement data is done in other applications, GEMS has to check these systems very often for new data. Increasing this load should only be implemented when a) the can be done very quickly and b) the chance of successful data retrieval is high. When the import takes more than 0.05 seconds or it could take days before a measurement is imported, these measurements should not be imported automatically, but the import should be initiated manually on a user request, e.g. pressing an import button.
A probably better solution is to have the external application initiate the import of measurement data. As GEMS is a web application this is easy to implement as all the exporting application has to do is call an import URL.

Data setup

The precise format of the patient and measurement data varies between implementations, but this only concerns the outward appearance of the data. Some projects show a birthday or use the home address of the patients; others want to know e.g. the physician or relatives of the patient. The data stored for a measurement depends on the type of measurement and changes even within a project.
What does not change is that all GEMS projects use internal patient keys that link the patient data to the measurement data and tokens that uniquely identify a certain measurement for a certain patient. I.e. a patient may have had 3 blood measurements, each of which has a different token that identifies the measurement values, the patient and the time of measurement.
Even the way the internal patient keys and the tokens are generated can be adjusted per project, but the existence of these identifiers does not change and these keys are always generated by GEMS.

The internal patient key is private to GEMS so should not be used by external applications. This is usually no problem as GEMS can also store other identifiers for the patient, e.g. social security number and/or patient number. When the patient information is retrieved on demand from an EPD this number is the only information needed to start the retrieval.
All imports and exports of patient data uses these keys and not the internal GEMS keys.

The token identifying a certain measurement for a patient is public. Tokens can be mailed to patients or can be handed out to them on paper.
To import a measurement the token generated by GEMS must be provided by the external application. In some research projects with a fixed number of measurements per patient it may be possible to match the measurements based on a patient identification and the order of measurements in time, but this approach is error prone and when a patient can have many measurements assigned by different people – as in a hospital setting – this is impossible.
GEMS can provide the tokens with a patient ID to the external application, so one solution is to export these tokens and then to return the measurement with the token. A second possibility is to have the token entered manually in the exporting application. The other possibility is to manually match each measurement to a corresponding token, e.g. using an import button linked to a token.

The only other requirement for data exchange is in the structure of the data provided: All information in GEMS should be provided as database tables. The possibilities for data import are therefore limited to structured formats like XML, CSV, EDIFACT, live database connections, etc…

The export has of course no such restrictions. Exports in PDF, Excel or Word format are possible, as well as the structured options mentioned above.

Communication setup

As GEMS communicates privacy sensitive data all communications will have to be secure, either through encryption or through limited network access. For remote GEMS installations this requires using VPN, SSH, SFTP or HTTPS for communication.

GEMS versus the Dutch National EPD

Some requirements for GEMS are fairly similar to the requirements for the Dutch National EPD as proposed by Nictiz. The primary form of communication will therefore be the same as that used by AORTA: HL7 version 3 in XML format exchanged using SOAP over HTTPS.

More information about AORTA can be found at the Nictiz site:
http://www.nictiz.nl/page/Landelijke-infrastructuur/AORTA.

One difference between AORTA and GEMS data exchange is that the Dutch EPD system is designed to exchange information about Dutch people only, while GEMS may be used for international research projects and even national projects may include foreign patients.

Another difference is that GEMS will need to exchange data in bulk for multiple patients or multiple measurements at once. Therefore the API for GEMS data exchange will differ a bit from the AORTA API, but the design goal is to make any AORTA implementation useable as a minimal GEMS online patient lookup implementation so that hospitals using an AORTA compatible EPD system can integrate with GEMS with minimal effort.
Projects where AORTA style communication suffices may set up their data exchange without any project specific programming at all.

Other options for GEMS data exchange

GEMS is an Open Source application written in PHP with a modular object oriented design. The requirements change with each GEMS project, so the code should be and is easy to adapt on a per project basis using a plug-in architecture.
The philosophy behind the architecture of GEMS is that while it may be impossible to write code that covers all possible projects, it is comparatively easy to write code that requires minimal effort to extend.

For data exchange GEMS will use a group of interfaces and abstract classes for on-line patient lookup and for bulk exchange, with some general plug-ins. E.g. to retrieve patient data from AORTA using a social security number will be a matter of providing a standard object class with

the security and identifying information needed. No actual code is required and as soon as the number is entered GEMS will search for the patient information.
Using HL7 over SOAP will require more information, but still does not require coding.
For bulk updates GEMS will provide similar standard modules working both over HTTPS and/or SFTP and for the non-secure versions of these protocols, using CSV, tabbed ASCII and HL7 (version 3 possibly even version 2.4) as the data format.
These standard modules will also be maintained by the GEMS project and new features will be added to them.

More specialized communication modules are also possible but do require coding. As the GEMS side of the communication is provided by existing PHP objects, the main effort required is writing the code that performs the specialized data retrieval. The cost in development time will only be for writing that code and depends only on the complexity of the communication.
PHP is known for it's high productivity and it's broad range of tools, including remote communications tools for COM, FTP, ODBC, pipes, RPC's, SOAP, XML, etc… so almost every tool in the programmer's arsenal can be used. So the costs need not be prohibitive, but will fall entirely on the project needing them. As will the costs for maintenance.