Flexible Database Platform for Biomedical Research with Multiple User Interfaces and a Universal Query Engine

Margus JÄGER[a,b], Liina KAMM[a,b], Darja KRUSHEVSKAJA[a,b], Harry-Anton TALVIK[b], Janno VELDEMANN[a,b], Andres VILGOTA[b] and Jaak VILO[a,b]
[a]*Institute of Computer Science, University of Tartu, Estonia*
[b]*Quretec Ltd, Tartu, Estonia*

Abstract. Biomedical research on human subjects often requires a large amount of data to be collected by personal interviews, Internet based questionnaires, lab measurements or by extracting data from paper or electronic health records. This data needs to be stored, analyzed and interpreted in a comprehensive manner. There is a great need for user-friendly software systems that allow to design and launch such data collection initiatives quickly, securely and reliably. We have designed a flexible software system that enables creating such data collection and management projects. Qure Data Management platform described in this paper contains tools for dynamic designing of data models and electronic questionnaires, multiple user interfaces for data entry and a universal query engine for filtered output. The platform enables to make database and questionnaire changes on the fly without any manual database table change. The data repository is based on the entity-attribute-value with classes and relationships (EAV/CR) approach which allows flexible storage and analysis of data from multiple studies. Although aimed at biomedical research, the software architecture is flexible and can be used for various different data collection and management projects.

Keywords. data management, questionnaires, entity-attribute-value, biomedical studies

Introduction

Data collection is essential for biomedical studies when gathering information in the form of questionnaires or clinical analysis results. Furthermore, it is important that analysis of the data collected can be performed. Hence, it is advisable to gather the data electronically. Usually studies last for a limited time and in practice they start rather abruptly therefore leaving little time to prepare the information system to back up the data collection.

New software systems are being developed even for individual research study projects [1]. In this paper a general system and architecture – the Qure Data Management™ platform – that allows many different research projects to benefit from the common research infrastructure is proposed. The developed system allows quick design of new surveys, hosting many simultaneous surveys on the same server

hardware, and integration with statistical software using a user-friendly query and data export system that would work even on very large data models consisting of up to thousands of attributes. One key feature of the system is ability to work offline without needing continuous Internet access that most today's web-based applications need.

The development of Qure Data Management platform began from designing the software solutions for the needs of the Estonian Genome Project [2]. This large-scale data collection activity to establish a new large-scale biobank required a software system to enable country-wide data collection. While designing the software it became evident that the required system had to be flexible enough to support an ever-changing set of questionnaire requirements and also to support a large number of parallel on-going studies.

The system has evolved since then. Now, Qure Data Management platform consists of a study design application Qure Designer, data entry applications – the desktop based Qure Desktop and web-based Qure Browser, a central data repository and communication server Qure Server, and a query engine Qure DataView.

Qure Designer is a graphical application for study design. It allows the user to prepare, preview and upload the data gathering environment. Qure Desktop and its web-based counterpart Qure Browser allow the user to enter data into previously designed views. The key benefit is that different user interfaces operate in essence on the same central database server using the specially designed synchronization protocol exchanging encrypted XML messages from certified parties.

The underlying data repository in Qure Server is based on the entity-attribute-value model with classes and relationships (EAV/CR) [3, 4, 5, 6, 7]. This model provides the means for easily making changes to the structure of the study, thus making it simple to add new attributes or whole perspectives when new requirements arise, without changing the underlying database. The query engine Qure DataView allows the user to retrieve data for analysis from the hierarchical EAV/CR database.

Even though flexibility and dynamism through EAV/CR have their drawbacks – large overhead, inefficient data retrieval – this model has proved feasible in short term studies. Qure Data Management platform has successfully been used in clinical trials to track subject visits and make systematic monitoring reports with well defined data structure.

The Qure Data Management platform is used for the Estonian cohort data entry of the study "Concerted Action on SeroConversion to AIDS and Death in Europe" [8]. The structure of data in the study is based on the HIV Cohorts Data Exchange Protocol standards [9], but because of the flexibility of the platform, it was possible to add new data fields for local research without remarkable work effort. It is also used in the study of Colon and Breast Cancer Diagnostics (COBRED) [10] to capture both clinical and logistics data. The largest current survey being performed on the Qure Data Management platform has over 70 object types and almost a thousand attributes.

The system can with small variations be used for Computer-Aided Personal Interviewing (CAPI), Computer-Aided Self-Administered Interviewing (CASI), or Computer-Aided Telephone Interviewing (CATI).

1. Qure Data Management Platform

Qure Data Management platform is used for collection, handling, and analysis of well-structured data under the requirements for high quality, security and robustness. The

platform incorporates tools for rapid designing of data models, user interface texts and entry form controls. End-user GUI is automatically created for an independent Windows client and web. The desktop clients and web interface can be used interchangeably, because they are built based on the same data model and views. The collected data is also shared and synchronized between clients. The user friendly Qure Desktop can even work off-line, fetching automatic updates and synchronizing data with the server as soon as the Internet connection becomes available. The description of the different parts of the platform and their relationships can be seen on Figure 1.

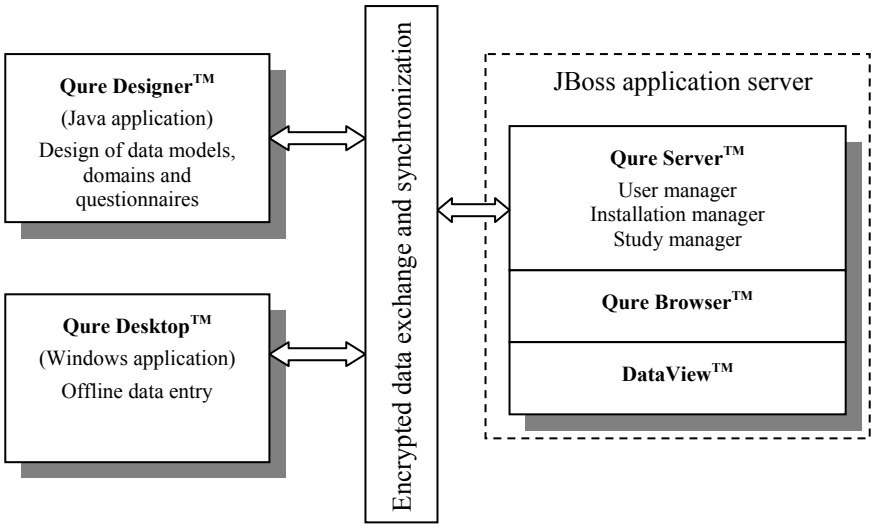

Figure 1. Qure Data Management platform

2. Central Data Repository

Qure Server consists of a central data storage engine and administration tools. Specialized web-based tools for data input, output, integration and queries are also integrated into to the server. The server implementation is based on J2EE technology and utilizes open source JBoss application server. The data storage technology is based on the EAV/CR architecture. An open source PostgreSQL database engine is used as a backend storage for the universal EAV/CR model (Figure 2).

The data is synchronized via an efficient synchronization protocol that propagates only the changes made to the entities, not the entities themselves. The synchronization messages are also compressed and encrypted to further minimize network utilization.

Security features of the server include certificate- and password-based authentication, single sign-on policy and role-based user management. The designer of the study can manage the roles, allowing differentiating the client access to study data. Restricting access to entities based on the entity creator is also possible.

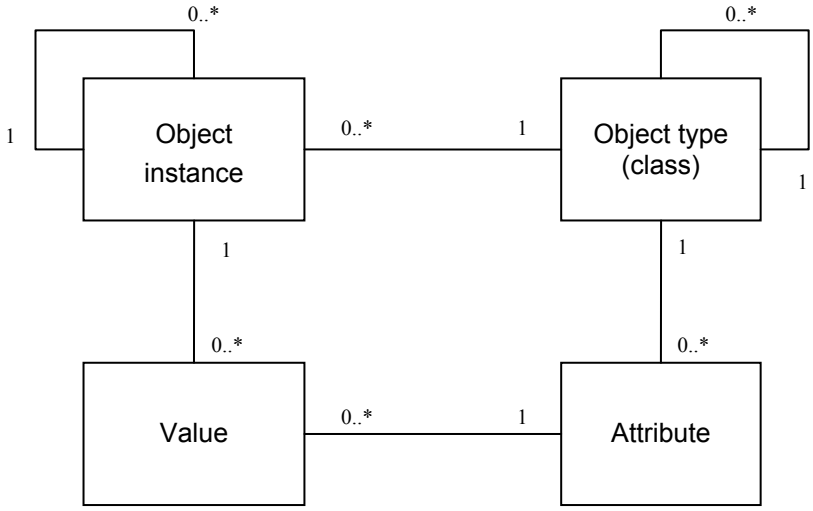

Figure 2. EAV/CR implementation

3. Data Model and User Interface Design

The data model is the backbone of a study. It is a hierarchical collection of object types and attributes definitions. In the context of the EAV/CR model, the term object type describes a class. Every object type contains attributes and may additionally contain other object types. Each attribute has a data type (eg. string, integer, date). Object types usually have one-to-many relationships, although if an entity has many different attributes, it becomes more difficult to find the necessary ones when for example trying to get output from the collected data. In that case the one-to-one relationship can be used for grouping purposes.

Domains are hierarchical classifiers or lists of items. Every item has a code and a label and may have sub items. Domains can be assigned to attributes and are usually used for one choice selection style data entry. Domains can also be assigned to object types – this possibility is mostly used for multiple value selections. The platform supports large hierarchical classification systems like ICD-10 [11] or standard geographical classifications.

User interface (view) is defined as a questionnaire. A questionnaire has multiple pages, every page contains question modules. Question modules are placed sequentially and every question module is connected to one attribute or object type. This connection defines where the data is stored. There can be multiple questionnaires based on the same data model covering different aspects for different users.

4. Pivoting Converter

Qure Server includes a tool for data integration with other systems. The data in entity-attribute-value storage model can be converted to the more conventional relational storage model. Introducing the pivoting converter allows us to harness the power of both models: internally we use EAV/CR model with metadata to support rapid model changes and synchronize data efficiently, but at the same time offer the query power and simplicity of relational databases to external clients [12].

In pivoting conversion every object type is transferred to a table in the relational database and attributes are transferred to table columns. In both cases the names from the object model are also converted respectively. As the object type names are unique in the data model scope and the attribute names are unique in the object type scope, there will be no naming conflicts. All tables in the relational database have primary key column "id" that contains the same object identifiers as in the entity-attribute-value storage. Relations with other objects are translated using the foreign key column naming convention (referred table name with the suffix "_fk").

The conversion process can be performed manually, but the server can be set up to convert incoming changes automatically, thus enabling us to make different reports and special applications customized according to user needs.

5. Designer Tool

The graphical tool Qure Designer can be used for designing object models, domains and questionnaires. It is a Java desktop application that stores all definition files in XML format.

The main purpose of the application is to give a more intuitive and usable interface to the user for designing study project files. As the study files are being validated in-place, the feedback is direct and the user can quickly fix the validation problems. The object model describes the object types and attributes that will later hold the data. Questionnaires define the graphical user interface for the object model – they present the attributes as questions and enable a more intuitive data entry environment. Questionnaire design is depicted on Figure 3. Qure designer also provides the user with a preview of the designed questionnaire.

The structure of a questionnaire can be changed quickly by dragging and dropping modules and the attributes can be modified by specific property fields. Questionnaires can be internationalized by translating question texts and domain item labels, thus, creating user interfaces in different languages.

The application provides the user with built-in searchable help system. With the help of Qure Designer the study files can be securely transferred to Qure Server.

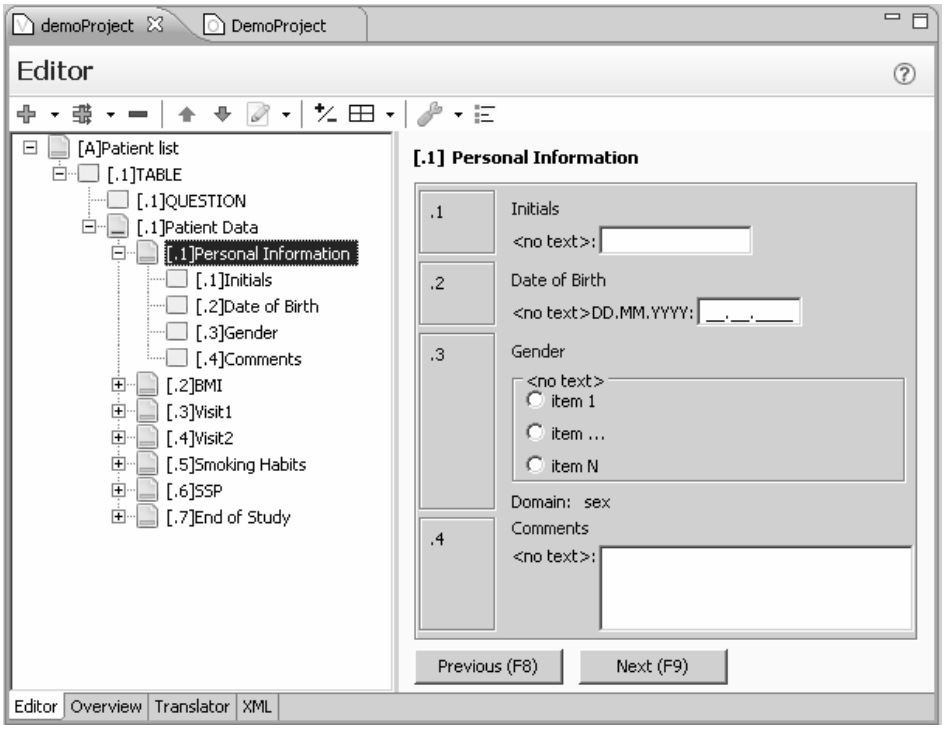

Figure 3. Questionnaire design

6. Online and Offline Data Entry

Qure Data Management platform has two user interfaces for data entry: Qure Browser for online usage and autonomous desktop client Qure Desktop for usage with low or no network connectivity. Qure Desktop (seen on Figure 4) requires Windows XP or Vista. Qure Browser (Figure 5) is integrated with Qure Server and is available to all web-enabled clients (tested with Firefox and Internet Explorer).

In clinical trials offline data entry support has been very important in conditions where Internet access is not always available [13]. For example CRA-s need to visit hospitals where Internet connection may not be available (for technical or security reasons). With offline support they can still access their study data in laptop computer and enter necessary information that can be synchronized later. In many cases Internet connection is just too slow or unstable that would make web-based or Internet dependent applications very hard and time consuming to use. In some cases computer-aided personal interviewing (CAPI) must be done, where an offline fat-client tool has a clear advantage in speed and stability.

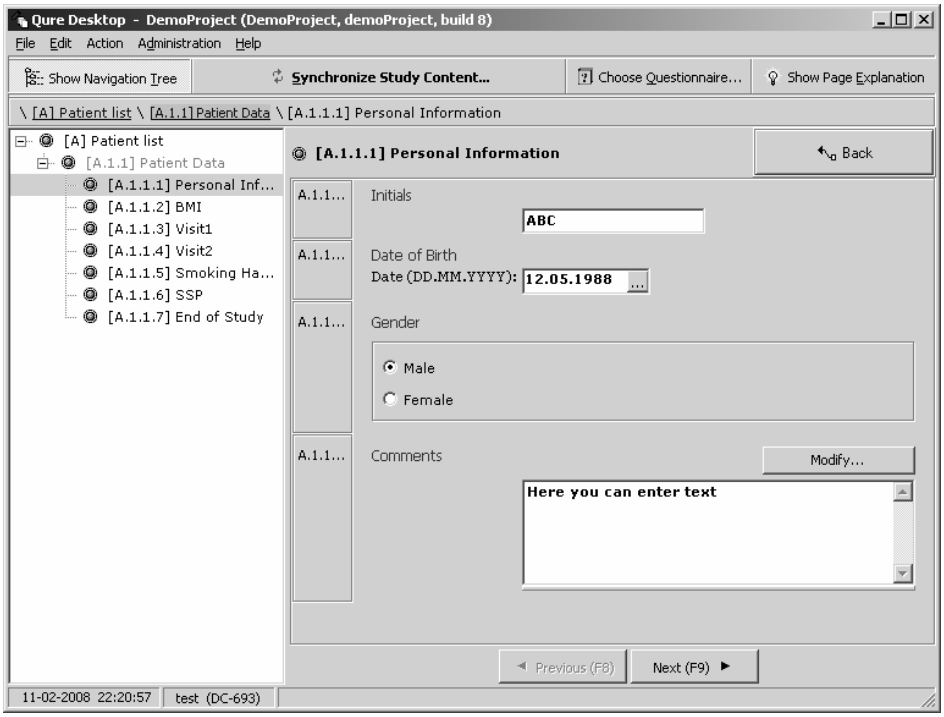

Figure 4. Qure Desktop user interface

Qure Desktop is an offline data entry tool that provides this possibility. In offline mode, the Qure Desktop caches the user changes in a local database and synchronizes the changes to the server when the Internet connection becomes available. While connected to the server, the user can also manually request data synchronization by one click of a button. The desktop client application is easily upgradeable to a next version via Software Update service.

The Qure Browser web tool is essentially an on-line version of Qure Desktop application, integrated to the server. The obvious benefit of Qure Browser is the ease of usage, the user does not have to download or install a specialized application. The user interface is generated from the same view definition and looks similar to Qure Desktop UI (Figure 5).

Data entry clients allow entering data to highly structurized model. The data quality and integrity can be improved by using various UI controls with pre-defined sets of answers, pattern-validated answers and even custom validation logic with dependencies to previous answers. For example, after asking persons age, a warning message can be displayed if a child's height does not fall within reasonable limits of persons age-group. Computed values can also be shown as default answers to questions.

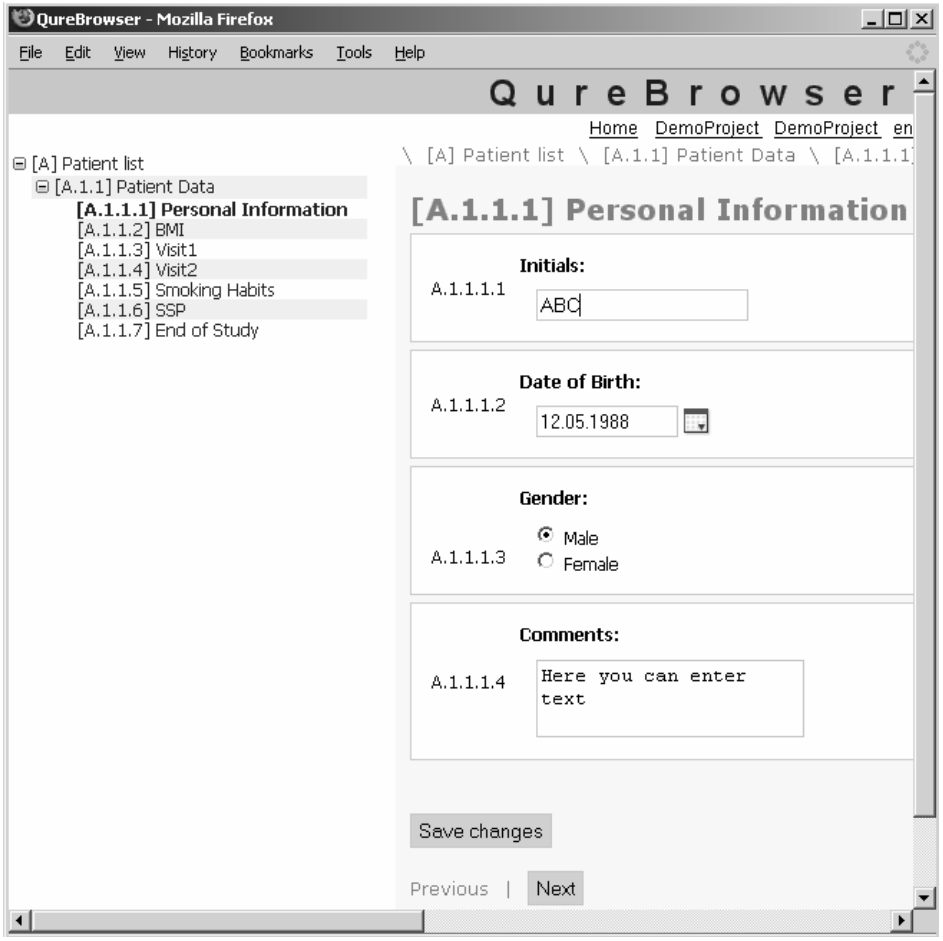

Figure 5. Qure Browser user interface

7. Universal Query Engine

Qure DataView is part of the web-based user interface of the central data repository. This tool enables the user to extract data from the Qure Server universal database fairly easy. The application takes the latest data from the database and transforms it into one flat dataset in the desired form (eg HTML, comma-separated values, XML), thus making it possible for the user to view the data or further analyze it in special data analysis tools. The main purpose of this tool is to extract data from the hierarchical database, and even though it is possible to carry out some basic filtering, aggregation and sorting, it is preferable to do the more complex analysis with tools that are specially designed for such a task.

As performance is a well-known problem [12, 14, 15] for systems with EAV data structure, a solution should be found for querying large amounts of data. To solve this problem, the inherent pivoting converter is put to use. If the data has been transferred to

a relational database for one study, then, for that study, the queries are done on the relational database, using the power of the query engine of the database. That gives us a significant performance increase on large data sets.

The user can specify the needed query parameters in five easy steps. Firstly, it is necessary to choose the main object type from the data model of the study. This parameter determines which type of entity will be in each row of the resulting flat dataset. There is no need to define joins between objects manually because the hierarchical data structure already defines parent-child relations between objects and joins are done automatically.

Secondly it is possible to define certain basic filters for attributes of the entity so the output will hold only a necessary selection of entities. Thirdly the user can specify which attributes are needed in the output (Figure 6). The query engine is also capable of making some simple aggregations if necessary. Finally the user can determine the sorting parameters and the format of the output.

In addition, it is possible to save the specified query parameters at any time, so that the user can later rerun the queries on the most recent data.

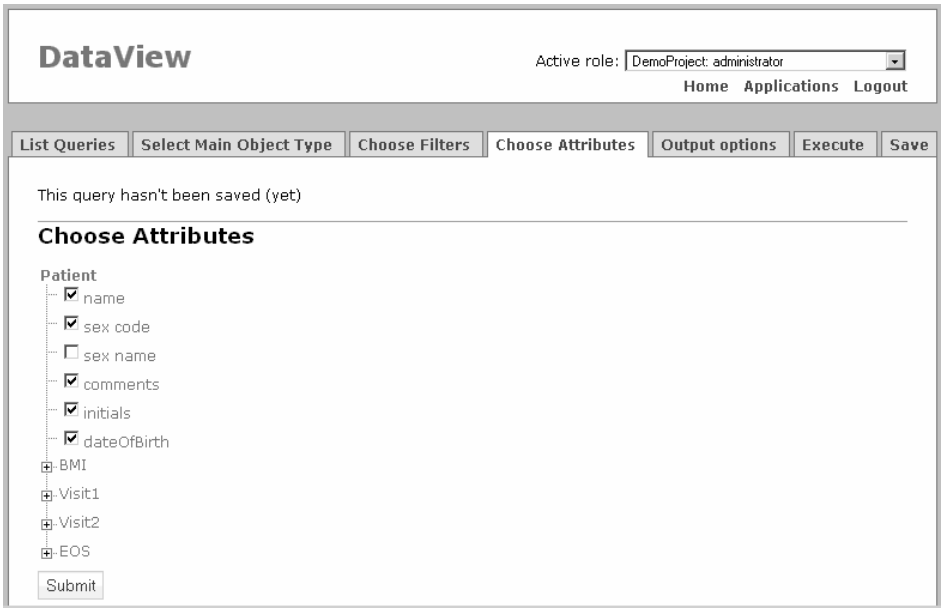

Figure 6. Selecting attributes in Qure DataView

Conclusion and Further Work

Qure Data Management platform is used for collection, handling and analysis of well-structured data under the requirements for high quality, security and robustness. The platform enables to make database and questionnaire changes on the fly without any database table structure changes that makes it suitable for biomedical studies where database changes are often a common requirement. Offline data entry support provided

by the platform is an important feature where Internet connection is not always available or UI response speed and stability are needed. The system contains a query engine with a friendly web based user interface. Performance issues with EAV structure are solved using a pivoting conversion to a relational database, where queries can be performed more efficiently.

The platform, though already in use, is a work in progress. We are working on optimizing both the desktop client and the server. When this is finished, we plan to do further performance analysis on different parts of the platform. At present, we intend to write separate articles about the query engine and asynchronous user rights management on our hierarchical data model.

Acknowledgements

This work has been supported by Enterprise Estonia project EU21289.

References

[1] Edwards, RL., Edwards, SL., Bryner, J., Cunningham, K., Rogers, A., & Slattery, ML. (2008) A computer-assisted data collection system for use in a multicenter study of American Indians and Alaska Natives: SCAPES, Computer Methods and Programs in Biomedicine, 90(1): 38-55.

[2] Metspalu, A., Köhler, F., Laschinski, G., Ganten, D., Roots, I. (2004) The Estonian Genome Project in the context of European genome research, Dtsch Med Wochenschr, Apr 30;129 Suppl 1:25-8.

[3] Nadkarni, PM., Marencol, L., Chen, R., Skoufos, E., Shepherd, G., & P. Miller. (1999) Organization of heterogeneous scientific data using the EAV/CR representation, Journal of the American Medical Informatics Association, 6(6): 478-493.

[4] Ganslandt, T., Mueller, M., & Krieglstein, F. (1999) A flexible repository for clinical trial data based on an entity-attribute-value model, Proceedings of the AMIA Symposium, 1999: 1064-1067.

[5] Los, RK., van Ginneken, AM., de Wilde, M., & van der Lei, J. (2004) OpenSDE: row modeling applied to generic structured data entry, Journal of the American Medical Informatics Association, 11: 162–165.

[6] Nadkarni, PM., Brandt, C., Frawley, S., Sayward, FG., Einbinder, R., Zelterman, D., Schacter, L., & Miller PL. (1998) Managing attribute-value clinical trials data using the ACT/DB client-server database system, Journal of the American Medical Informatics Association, 5(2): 139–51.

[7] Nadkarni, PM., Brandt, C. (1998) Data extraction and ad hoc query of an entity-attribute-value database, Journal of the American Medical Informatics Association, 5(6): 511–27.

[8] CASCADE: Concerted Action on SeroConversion to AIDS and Death in Europe (available at http://www.ctu.mrc.ac.uk/cascade/)

[9] HICDEP: HIV Cohorts Data Exchange Protocol (available at http://www.cphiv.dk/Portals/_default/pdf_folder/HICDEP.pdf)

[10] COBRED: Colon and Breast Cancer Diagnostics (available at http://www.cobred.eu/)

[11] International Classification of Diseases (available at http://www.who.int/classifications/icd/en/)

[12] Beck, P., Truskaller, T., Rakovac, I., Cadonna, B., & Pieber, TR. (2006) On-the-fly form generation and on-line metadata configuration--a clinical data management Web infrastructure in Java, Studies in Health Technology and Informatics, 124: 271-276.

[13] Fraser, HS., Jazayeri, D., Nevil, P., Karacaoglu, Y., Farmer, PE., Lyon, E., Fawzi, M.K., Leandre, F., Choi, SS., & Mukherjee JS. (2004) An information system and medical record to support HIV treatment in rural Haiti, British Medical Journal, 329(7475): 1142–1146.

[14] Marenco, L., Tosches, N., Crasto, C., Shepherd, G., Miller, PL., & Nadkarni, PM. (2003) Achieving evolvable web-database bioscience applications using the EAV/CR framework: recent advances, Journal of the American Medical Informatics Association, 10(5): 444-453.

[15] Chen, RS., Nadkarni, PM., Marenco, L., Levin, F., Erdos, J., & Miller, PL. (2000) Exploring performance issues for a clinical database organized using an entity-attribute-value representation, Journal of the American Medical Informatics Association, Sep–Oct; 7(5): 475–487.