BIO-ALGORITHMS AND MED-SYSTEMS
JOURNAL EDITED BY MEDICAL COLLEGE – JAGIELLONIAN UNIVERSITY
Vol. 5, No. 10, 2009, pp. 59–69

HEARTFAID'S ECRF: LESSONS LEARNT FROM USING A TWO-LEVEL DATA ACQUISITION AND STORAGE SYSTEM FOR KNOWLEDGE DISCOVERY TASKS WITHIN AN ELECTRONIC PLATFORM FOR MANAGING HEART FAILURE PATIENTS

[1] Andrzej A. Kononowicz, [1] Katarzyna Styczkiewicz, [1] Bogumiła Bacior,
[2] Matko Bošnjak, [2] Rajko Horvat, [2] Marin Prcela, [2] Dragan Gamberger,
[3] Angela Sciacqua, [5] Maria Consuelo Valentini, [1] Kalina Kawecka-Jaszcz,
[4,5] Gianfranco Parati, [6] Domenico Conforti

[1] Jagiellonian University Medical College, Kraków, Poland
[2] Rudjer Boskovic Institute, Zagreb, Croatia
[3] University "Magna Graecia" of Catanzaro, Department of Experimental and Clinical Medicine, Italy
[4] University of Milan – Bicocca, Department of Clinical Medicine and Prevention, Milan, Italy
[5] Department of Cardiology, S.Luca Hospital, Istituto Auxologico Italiano, Milano, Italy
[6] University of Calabria, Department of Electronics, Informatics, Systems (DEIS), Italy

Abstract: Case report forms are important sources of medical knowledge in all clinical studies. Electronic versions of these forms have several advantages compared to traditional paper-based questionnaires, and they have been adopted in many contemporary research projects in medicine. This paper presents a framework for creating case report forms designed with a two-level approach. Data at the generic information model level is stored in EAV (entity-attribute-value) tables and extended by tables facilitating specification of the questionnaire layout. The second layer (knowledge model) specifies the domain specific concepts describing the field of application of the questionnaire. This framework has been applied and tested in the frame of an EU FP6 research project – HEARTFAID – the objective of which was to build a knowledge-based platform supporting the management of elderly patients suffering from heart failure. Data collected by the electronic case report form (eCRF) was used in the project's knowledge discovery and decision support tasks. The work presents a new way for effective extraction of the data necessary for the integration with the knowledge discovery process in a distributed, service oriented framework of the HEARTFAID platform. It is demonstrated that it is feasible to implement these tasks using the two-level EAV table design.
Keywords: Electronic Data Capture, Remote Data Entry, EAV, Two Layers Modelling, eCRF, Spring Framework

1. Introduction

Electronic Data Capture (EDC) techniques have been used in clinical trials for a long time [8]. The first EDC systems (also known under other names e.g. Remote Data Entry (RDE) Systems) date back to the early 1970s [7]. Since that time a huge amount of applications (either academic or commercial) for creating, managing and publishing medical on-line forms has been developed. The electronic versions of questionnaires seem to have lot of advantages in comparison to their paper-based counterparts [18]. Among the assets of EDC are cost savings, faster dissemination of forms and collection of data, built-in validation mechanisms, easy maintenance and export to statistical packages.

The conventional method of designing database schemes for questionnaires is to map a form to a single table or a set of tables in a relational database in which each attribute (question from the form) is stored into an individual column [10, 14]. Even though this technique works fine for many applications, it has become apparent that this method is not always effective [14], especially in bio-medical research or electronic health records. This problem pertains to databases with a large, heterogeneous list of fields from which many are optional and can be omitted. In such databases new fields are often added, altered or removed after the database has been deployed, and this introduces additional complication in its structure. Designing such databases with the conventional approach is possible but often troublesome and ineffective due to the limitations of traditional RDBM systems (e.g. a maximum limit of 255 columns in a single database table in some RDBM systems) or to the need to frequently update the database structure.

An alternative approach is to store records as association lists containing *(attribute name, attribute value)* pairs of variables [13]. A database that stores information in that form is called an entity-attribute-value (EAV) database. This storage method by itself is not new since it dates back to at least the time when the LISP programming language was created. However, its application in relational databases has not yet been very widespread. Classical EAV-databases contain one large table with just three columns: identifier of the described object, identifier of the attribute, value of the attribute. Additionally, dictionaries are required which contain metadata describing the attributes applied.

This simple design technique enables a very flexible method of space-efficient storage of heterogeneous data. However, it should be acknowledged that also this approach is not free from flaws. It is well suited for one object-at-a-time queries in which all information about a single object (e.g. patient) is returned, but it is less efficient in complex attribute-centric queries [14]. For such situations special frameworks facilitating more advanced searches in this model are implemented (as e.g. QAV: *querying entity-attribute* framework [13]), thus empowering the user to browse the data more easily. The overhead that is needed to organise the data in an EAV manner is often not worth the effort for the simple and static databases used in many business applications. The queries are also not time-effective, rendering them less suitable for commercial usage. These drawbacks are, however, not as obvious in research projects and clinical trials where more emphasis is put on the flexibility of the tool than its efficiency.

The EAV model can be considered the first generic layer of a database. This tier may be used in virtually any field of application, and can be extended by additional tables supporting more complex data design. An example of a model with such additions is represented by the EAV/CR by Nadkarni et al. [14], which enhances the EAV by structures for the representation of classes and relationships. Other approaches customize the EAV to store clinical forms [5]. The EAV model with its extensions represents an information model of the database which is domain independent. In a two-level approach to database design, a second layer (i.e. the knowledge model), is added on top of the first [11]. This model specifies the domain specific concepts describing the field of application of the questionnaire. It may consist of terminologies and ontologies related to a given specialization field. The values that can be entered into the information model can be constrained by knowledge model archetypes [2] – i.e. special templates that specify at runtime the way data can be entered. Archetypes may be specified autonomously by subject matter experts (e.g. clinicians) without the need to consult database specialists. A clear separation between the first and second level of the database makes the architecture flexible and reusable.

The aim of this paper is to report on the information obtained while implementing a vast two-level electronic case report form (eCRF) which was designed for the cardiology domain. The eCRF is part of a large knowledge-based platform called HEARTFAID supporting the management of elderly heart failure patients, developed in the frame of the EU FP6 research program. It was required by the HEARTFAID project that the eCRF system implements insertion, modification and querying of large forms (containing over 700 attribute values for each patient). The system needed to be well integrated with the remaining services of the platform.

2. The HEARTFAID Platform

Heart failure (HF) occurs when the heart fails to pump enough blood to meet the metabolic needs of the body's tissues and/or organs. The prevalence of this pathological condition is very high – approximately 10 million patients suffering from HF in Europe. Chronic (C)HF is a disease of older people; the Framingham study noted a doubling of prevalence with each advancing decade, reaching a rate ranging from 7% to 10% in those aged 80 and older. The mortality of patients with severe HF is also high, approaching 50% in the course of one year in NYHA IV[1] class patients. However, it is believed that through regular monitoring and personalised management of patients affected by this condition, their survival rate and quality of life can be significantly improved.

The role of the HEARTFAID platform is to support physicians and healthcare personnel (e.g. nurses) in managing heart failure patients, while at the same time empowering patients to self-monitor their health condition [3, 4]. HEARTFAID is a web-based platform of services integrating several diverse modules (Fig. 1). Its basic function is to collect patient-related biomedical data from different sources (e.g. mobile and wearable measurement devices or medical imaging systems) and enable access to previously collated data from electronic health records. Part of the system includes declarative and procedural knowledge taken from evidence-based sources such as medical guidelines and carefully selected research papers [6, 16]. The users of the platform may securely access the data it contains in a standardised manner. The platform gives access to data taking into account the different roles and rights of the users. New knowledge can be discovered based on the data collected on the platform by employing newly developed artificial intelligence tools. The system has the potential to support physicians in making clinical decisions in the workplace also by alerting them if a dangerous situation is detected. All HEARTFAID services are integrated by an enterprise service bus (ESB). The system utilises a single sign-on mechanism. Users interact with the system through a customisable web portal. The anticipated results of integrating the platform into clinical practice include a reduction in the re-admission of HF patients to hospital, improvement in the quality of treatment and a decrease in management costs [3, 4].

A knowledge-based platform like HEARTFAID requires various forms of medical data acquired from different sources. Data collected from mobile and wearable devices are covered by the AmI service. The role of the HEARTFAID's electronic case report form (eCRF) is to handle all data required by clinicians that need to be inserted manually by the medical personnel. Beyond the scope of the eCRF is the storage of medical knowledge in the form of rules or ontologies which are used for inference in knowledge discovery and decision support systems. However, these modules exploit the data collected by the

1 NYHA – New York Heart Association Functional Classification – A four scale classification of heart failure extent

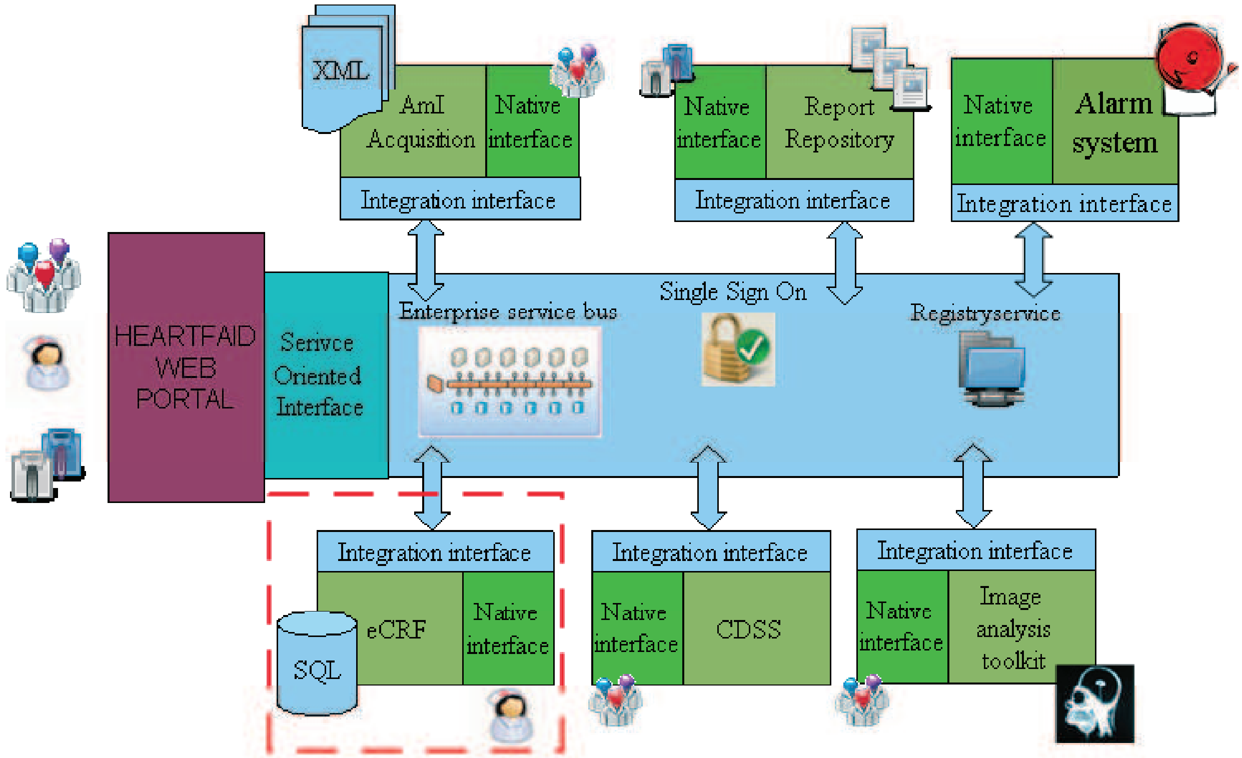


Fig. 1. General overview of the HEARTFAID services

eCRF. The eCRF is intended to be used by medical personnel in the hospital and is not accessible by patients. It plays the role of a specialised electronic health record, collecting heart failure data from a multitude of sources. From a medical (i.e. cardiographic) perspective the eCRF is useful because it gives easy access to the results of lab tests, to treatment schedules and to the prognostic assessment of HF patients.

The HEARTFAID eCRF comprises three parts: the baseline, additional visits and final evaluation forms. Each of these forms is uniquely assigned to a patient and can be filled out only once, with the exception of the additional visit form which may be repeatedly compiled without limitations. Questions in the eCRF questionnaire may be combined into groups. The activation of a group may be triggered in real-time by the input of a specific value by the user. Question groups may be nested to an unlimited depth. Most of the questions are of simple types: Boolean values, text strings, numerical values (integer, real numbers) and dates. However, there are also more complex types of questions which involve, for instance, the selection of a value from a controlled vocabulary, or the use of a special tool to specify a medication and its dosage from a hierarchical list of products (drug class, international name and generic name). It is also possible to add new drugs to the list. Some questions are grouped into matrices (tables) of values of simple types. The forms also contain rules for validating inserted values.

3. Method

While planning the implementation of the eCRF we looked for off-the-shelf products that were web-based, available free, open source, flexible enough to add new question types, able to support large questionnaires with nested groups of questions, based on XML and J2EE technologies, and easy to integrate into the HEARTFAID platform. None of the existing tools we found for designing web-based questionnaires (e.g. ArchiMed [5], Form Handler [21], Instant Survey [24], Survey Monkey [26], WebEAV [15], Zoomerang [27]), fully met our demands. For that reason it was decided to implement the eCRF from scratch. Since the number of questions was large (more then 700), quite diverse, potentially changeable and the efficiency of the tool was not a critical factor, it was decided to employ a two level architecture. The idea of a two-level approach emerged in electronic health records development [11]. Following this approach database structures are divided into two separated models: information model and knowledge model. The information model represents stabile and generic concepts, whereas the knowledge model depicts the dynamics of the problem field [11]. In the HEARTFAID's eCRF the information model expresses a generic database for storing clinical forms following the EAV paradigm. The classical EAV data model has been extended to facilitate the usage of complex web-based forms. In Fig. 2 the ERD (entity-relationship-diagram) of the information model underlying the HEARTFAID eCRF is presented. The model is generic – i.e. it does not contain any information specific to the heart failure domain and can be used in diverse multi-centred clinical trials. The EAV model was implemented in a RDBM system. Basic EAV tables were extended by additional tables for storing hierarchical question groups and for user management. Similar approach was taken in other EAV projects (as e.g. in EAV/CR representation by Nadkarni at al [14]). The *User Center* and *User* tables enable the separation of patients coming from different research institutions and enable access to the data only by entitled users. The *Patient*

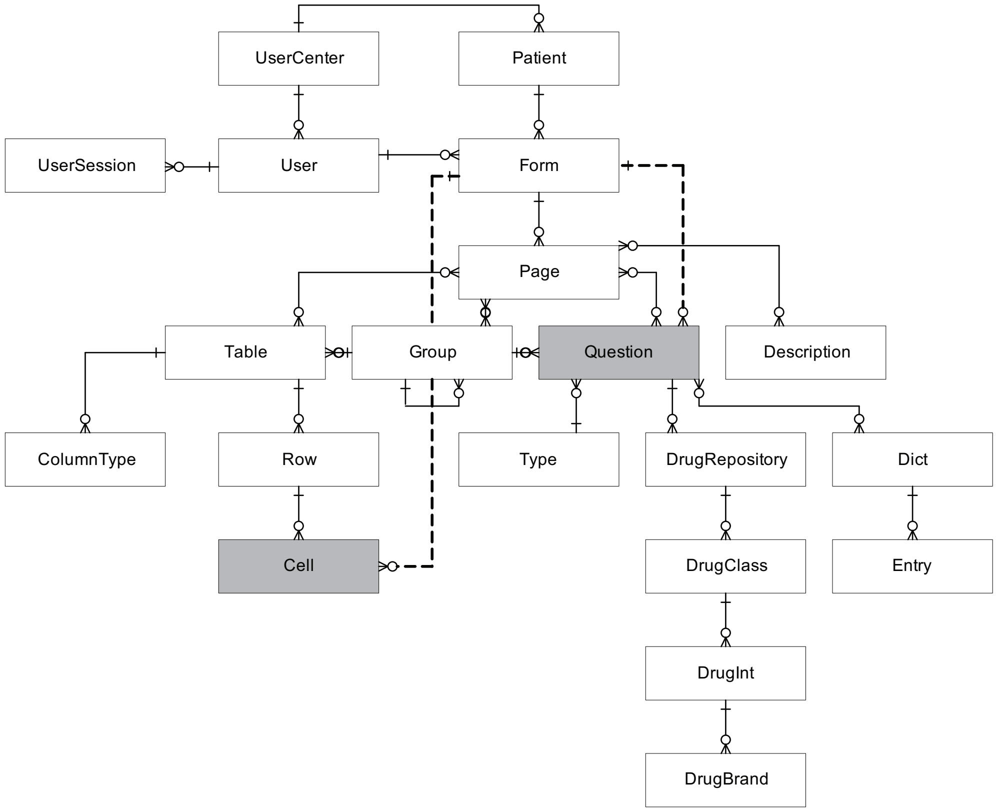


Fig. 2. ERD of the information model under laying the HEARTFAID eCRF

```
<bean id=”physical_exam_systolic_blood_pressure” class=”org.javs.ecrf.mvc.model.Type” singleton=”true”>
     <property name=”type” value=”integer”/>
     <property name=”html”>

     <value><![CDATA[Systolic blood pressure:]]></value>
     </property>
     <property name=”cui” value=”C1306620”/>
</bean>

<bean id=”question_hfcrfsa_1_g1_1” class=”org.javs.ecrf.mvc.model.Question” singleton=”false”>
     <property name=”order” value=”1”/>
     <property name=”type” ref=”physical_exam_systolic_blood_pressure”/>
     <property name=”value” value=””/>
</bean>
```

Fig. 3. XML archetypes specifying the values that can be inserted into the form

table contains basic patient data. Since the questions are assigned to pages and these pages may contain many levels of nested question groups or tables this structure is reflected by the *Page*, *Table* and *Group* entities. The grey-shaded *Question* and *Cell* tables are classic EAV tables containing a reference to the type of the question, the owning entity (i.e. *Page*, *Group*, or *Row*) and the value. Additionally, these tables contain information about the time of creation and last modification of the values, version number, as well as the identity of the user that modified the value. The dashed line between the *Question* or *Cell* tables and the *Form* table is a redundant connection added for efficiency reasons to accelerate queries with fields nested deep in many subgroups. The *Description* table contains textual information needed as additional description in the forms. The *Drug[X]* (*X*∈{*Repository,Class,Int,Brand*}) and *Dict* tables represent respectively the pharmacological treatment and values from controlled vocabularies.

The archetypes (second layer of the model) constraining the values that can be inserted into the database are specified in XML syntax, compatible with the bean definition syntax of the Spring Application Framework [9]. An example of the specification of a question type and its instance is presented in Fig. 3. The first archetype bean example defines a type representing a patient’s systolic blood pressure taken during a physical examination. This question has its description in HTML format (attribute *html*), stating that it accepts only integer values (attribute *type*) and a mapping to a concept in the UMLS ontology explaining its semantics (attribute *cui*). The second bean is an instantiation of the previously mentioned question type (attribute *type*). The position at which the question is displayed in the question group is specified by the attribute order, and its default value may be specified by the attribute value. The archetypes often also contain lists of questions or subgroups aggregated by group type, or information about question groups being activated or deactivated based on specific values of the questionnaire fields inserted by the user.

Both archetype beans (i.e. type declaration bean and question instantiation bean) are mapped to POJO (plain old java objects) elements and are stored on demand in the eCRF database using the Hibernate Framework [12]. The way the archetypes are specified enables easy extension of the list of constraining rules (e.g. by information about the soft or hard limits for ranges of accepted values).

4. Results

The eCRF has been implemented in the course of the second year of the HEARTFAID project in the Java 5 programming language. The development has been accelerated by the usage of the Spring Application Framework [9] version 1.2 and Hibernate 3 [12]. The final knowledge model of the eCRF specified by XML archetypes included 735 question instances of 364 semantic types. Archetypes were created using a general purpose XML editor (Altova XMLSpy 2008 [20]). Data were stored in a MySQL 5.1 RDBM system. A simplified structure of the eCRF is presented in the figures included in the Appendices 1 and 2. In order to make the schemes legible, the number of fields for each object was limited to a maximum of 10 fields. The letters *b,a* and/or *f* denote in which eCRF type of form this question is located (i.e. *baseline*, *additional form* or *final visit*). The forms are presented online as HTML views created with XSLT transformation of XML archetypes and data retrieved from the database (Fig. 4). The *top bar* contains the questionnaire's name, patient id and page number. The pages can be changed either through the list of pages in the *table of content* panel in the right part of the form or through the *backward* and *forward* buttons in the navigation bar. The form is automatically saved after changing a page or after clicking on the *submit* button. Activation of the *cancel* button rejects the last changes and exits the form. Question groups are marked by red boxes and activated by trigger questions (e.g. in Fig. 4 the group containing the *max. ST depression* question is activated by setting the value "yes" in the *ST depression* field). In Fig. 5 a 3x3 question table (matrix) of integers is presented. In addition, above the main form a list of detected validation errors is presented.

Communication with the eCRF with the HEARTFAID platform is established through an XML protocol implemented by one of the partners in the project (SYNAPSIS) including all the necessary information of an HL7 message [22] and following the transactions suggested by IHE [23]. The HEARTFAID middleware implements Patient Demographic Query HL7 V3 (PDQ) and Patient Identifier Cross-Reference HL7 V3 (PIX) profiles. In order to integrate the patient-related data into the platform a MPI (Master Patient Index) service is used which manages patient's demographic information and guarantees their unique identification in the environment. For instance, while registering a new patient on the platform, a message is sent from the HEARTFAID portal to the ESB which was implemented using the Mule open-source framework [25]. Mule descriptors for routing the information to a MIDA Graph (a workflow engine implemented by SYNAPSIS [19]) are read and transformed into information that is stored in the MPI and transmitted as HTTP XML messages to the eCRF service. The eCRF receives the messages, enrols the patient and sends back a confirmation message [19].

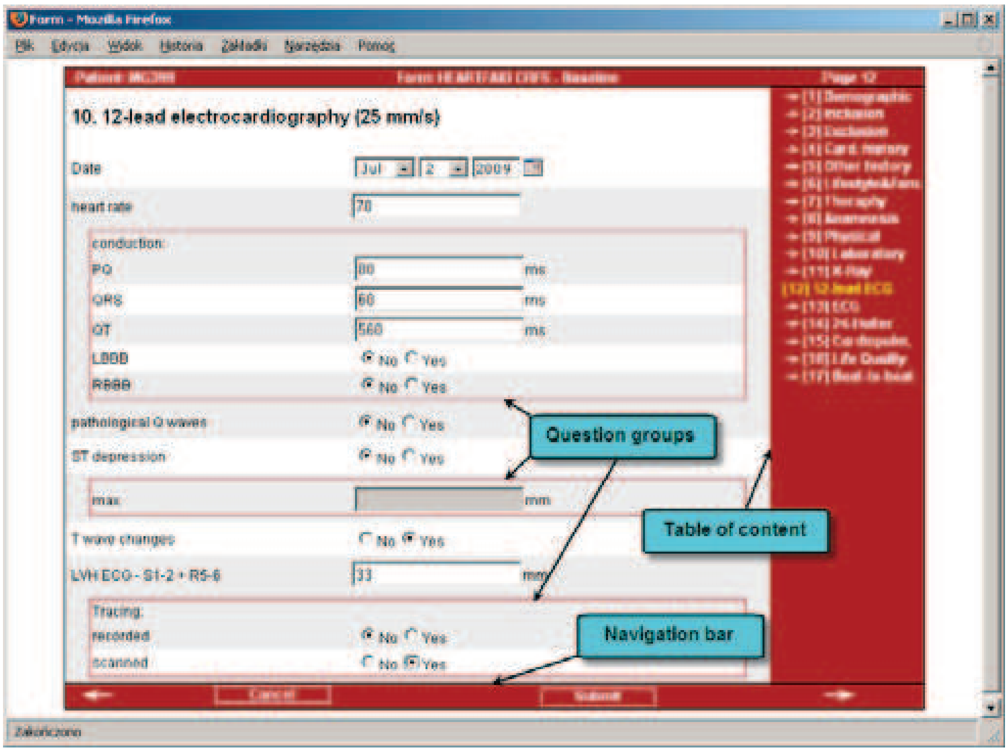


Fig. 4. User interface of the HEARTFAID eCRF

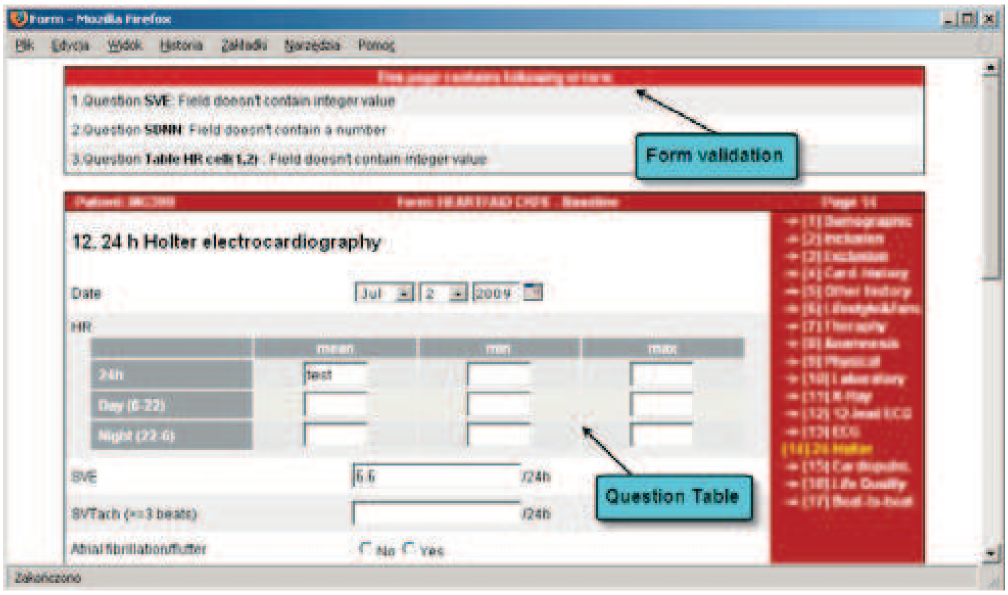


Fig. 5. Question tables and form validation in eCRF HEARTFAID

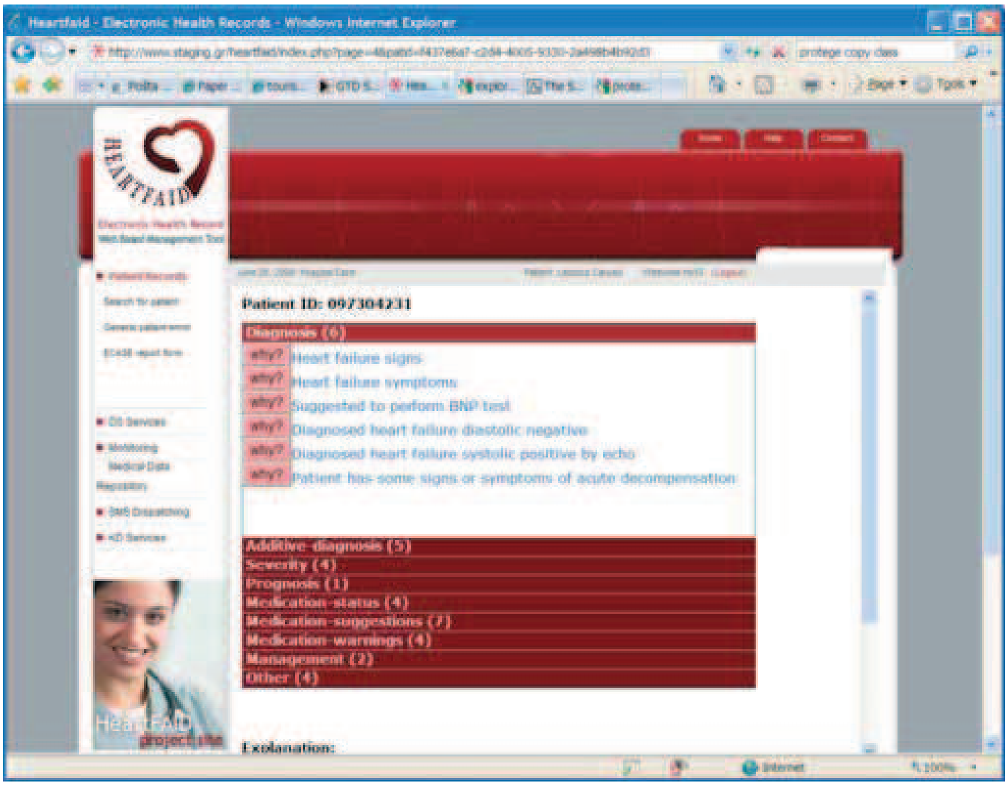


Fig. 6. The result of reasoning based on the data collected by the eCRF

The eCRF was deployed on the HEARTFAID platform in 2007 and since then it has been in constant use. Data from approximately 100 patients from four clinical centres [Università degli studi Magna Graecia, Catanzaro (Italy), Università degli studi di Milano Bicocca, Milan (Italy), Jagiellonian University Medical College, Kraków (Poland) and S. Luca Hospital, Istituto Auxologico Italiano, Milan (Italy)] have been collected.

The eCRF has been integrated with the HEARTFAID's Knowledge Discovery Service (KDS) and Decision Support System (DSS) developed by Rudjer Boskovic Institute in Zagreb (Croatia). Both services require tight integration with the large amount of patient data collected by eCRF, however these services require substantially different data access types. DSS is always focussed on one patient while KDS requires information about all or most of available patient data that has been collected by the eCRF. Additionally, it must be noted that DSS requires effective access to the most recent information for all potentially relevant measurements regardless of when they were collected and with a clear indication about when the data was acquired. In contrast to this, actual data collection time is not relevant for KDS, but it requires access to the data grouped according to the time of its collection, that data should be ordered by its historical order, and that it is identified by the time interval from previous measurements. Fig. 6 demonstrates a typical result from the decision support service while Fig. 7 and 8 illustrate the knowledge discovery service.

A unique property of the currently implemented KDS is that it integrates knowledge discovery algorithms with direct database access into one web-based service. This is not a simple task due to the complexity of the KD process [16]. The HEARTFAID service implements the modern random forest based machine learning algorithm [1] that has been reimplemented by Rudjer Boskovic Institute. The service has been built as a series of projects so that every project consists of different datasets with many tasks that can be performed for every dataset. Access to projects, datasets, and tasks is enabled though a web interface (Fig. 7).

Computationally, the most complex part of the service is the construction of the classifier and the preparation of a report (Fig. 8) based on the results of this process. The value of this newly implemented service is the realisation of direct access to the data in the eCRF and its automatic transformation into a form that can enter the KD process. Direct access to the relational database containing EAV tables by a traditional SQL interface is very laborious. This problem can be solved by implementing a special query module for external analytical services. An interface consisting of four generic functions has been implemented for the purpose of the knowledge discovery task. Table 1 contains a description of these functions:

These functions are available through a HTTP GET interface. In the following line a command is shown that starts a query involving the *getLastValue* function http://localhost:8080/heartfaid/query.html?function=getLastValue&uuid=312&sid=physical_exam.weight

After execution the interface searches in the eCRF database for all values of the *physical_exam.weight* attribute regarding the patient with the id 312. The result of the function is returned in simple XML syntax. This allows a clear separation of the data collection and query tool located at one centre (currently at Jagiellonian University Medical College, Kraków, Poland), from the knowledge discovery system located at a remote centre (currently located at Rudjer Boskovic Institute, Zagreb, Croatia).

Discussion

After the generic framework for designing questionnaires had been developed, the process of implementation of the HEARTFAID's eCRF knowledge model by specifying the XML archetypes took little time and did not cause any difficulties. The structure of the eCRF turned out to be more stable than initially anticipated, so the benefit of the flexibility of the architecture has not been fully used (with the exception of a few minor changes). On the other hand, the drawbacks of decreased database efficiency in this type of application are hardly noticeable. In the production database containing just a few users and approximately 100 forms installed on a Intel Core 2 Duo T5450 1,66Ghz,1GB RAM computer, loading a whole form from a database took in average 1360 ms, saving a modified

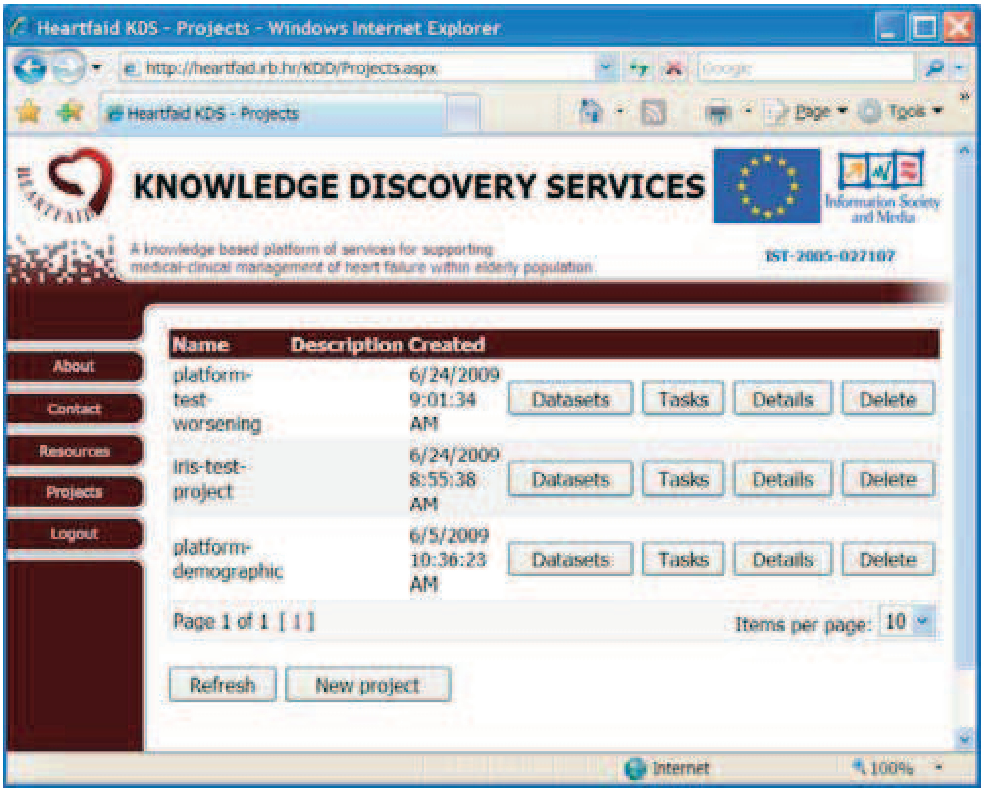
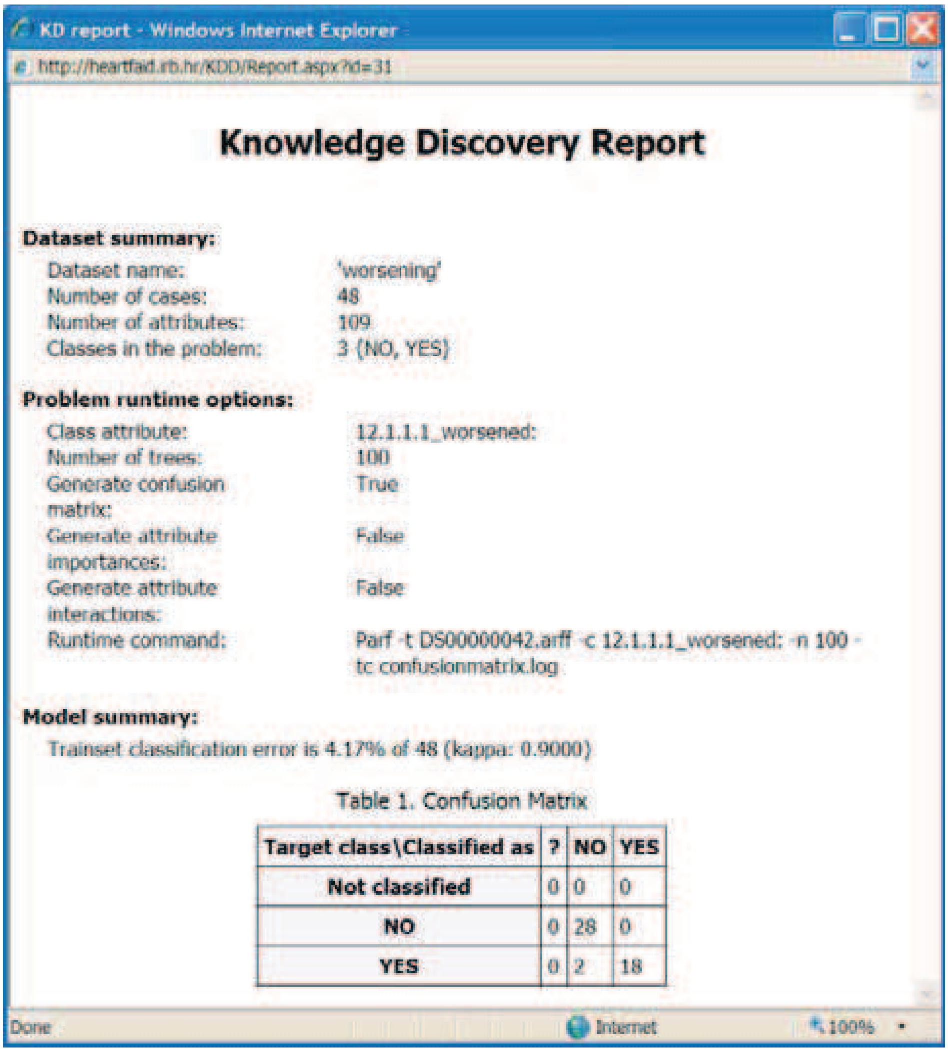


Fig. 7. The main page of the HEARTFAID knowledge discovery service with three current projects: "platform-test-worsening", "Iris-test-project", and "platform-demographic"



KD report - Windows Internet Explorer

http://heartfaid.irb.hr/KDD/Report.aspx?id=31

Knowledge Discovery Report

Dataset summary:

Dataset name: 'worsening'
Number of cases: 48
Number of attributes: 109
Classes in the problem: 3 (NO, YES)

Problem runtime options:

Class attribute: 12.1.1.1_worsened:
Number of trees: 100
Generate confusion matrix: True
Generate attribute importances: False
Generate attribute interactions: False
Runtime command: Parf -t DS00000042.arff -c 12.1.1.1_worsened: -n 100 -tc confusionmatrix.log

Model summary:

Trainset classification error is 4.17% of 48 (kappa: 0.9000)

Table 1. Confusion Matrix

Target class\Classified as	?	NO	YES
Not classified	0	0	0
NO	0	28	0
YES	0	2	18

Done Internet 100%

Fig. 8. The result of any KD task is a report. The figure presents a report for a two-class domain obtained after constructing a random forest with 100 trees. The main part of the report is the confusion matrix demonstrating the predictive accuracy measured by cross-validation on the training set.

form 600 ms, querying the last value of a selected parameter 84 ms. . Thanks to the application of XML technology the integration of the eCRF to the platform's enterprise service bus was easy and fulfilled the requirements of current medical informatics standards.

The future plan for the proposed architecture includes implementation of a graphical editor for the XML archetypes and extension of the list of constraints that can be used for the knowledge model's specification. Tighter integration of the eCRF with knowledge engineering and data mining tools through the proposed interface also seems to be important.

It is not easy to give definite advice about when to use EAV tables instead of traditional relational database design. If our highest priority is flexibility, and the number of collected attributes is very large and potentially often changeable, this suggests that a two-level EAV design should be used. In all other cases, a more traditional design would probably be more advantageous. When designing frameworks with EAV databases for knowledge discovery tasks it is imperative to also offer a special query module with an interface similar to that presented in this paper, or to export the data to an external system with a different data model.

Conclusions

This paper presents a practical implementation of a two-level database system for a medical research project. The generic layer of this database uses EAV tables which are useful for designing large heterogeneous and frequently changeable database schemas, as are often found in research studies. In this system a method for implementing the concept of two-level architecture in a modern application framework (Spring Framework) has been demonstrated. The fact that the system has been in use for almost two years in the HEARTFAID project and has delivered useful data for other modules like a knowledge discovery module and decision support services proves the feasibility and the effectiveness of this solution. The significance of our work consists in the proposal of a new type of direct interface for accessing complex data structures with the

Tab. 1. eCRF interface for knowledge discovery tasks

Function Name	Description
getLastValue	Returns the last known descriptor value available for the patient. If all values are unknown the returned value is also unknown.
getAnyValue	Returns information concerning all previous visits. For numerical measurements it returns two values: minimum and maximum while for categorical attributes it returns most frequent (mode) value. If all values are unknown the returned value is also unknown.
getDifference	Returns the difference between the last available piece of data and the penultimate piece. If there are not two available entries the value is unknown. For numerical attributes (e.g. laboratory values) it is the difference (+/- value). For categorical attributes it is 0 (no change) and 1 (value changes) [or -1 improved, 0 no change, 1 worsening]
getFlattenedTable	For categorical values it returns the number of known values and the most frequent value. For numerical it returns mean, minimal and maximum value, range, standard deviation and slope.

output already prepared for artificial intelligence applications. Additionally, it is also clearly stated that this model is not appropriate for every database, especially not for large commercial databases, and therefore its adoption needs to be carefully considered.

References

1. Breiman L., Random Forests, Machine Learning 45(1), pp. 5-32, 2001.
2. Bird L., Goodchild A., Tun Z., Experiences with a Two-Level Modelling Approach to Electronic Health Records, Journal of Research and Practice in Information Technology, 35(2), pp. 121-138, 2003.
3. Chiarugi F. et al., Support for the Medical-Clinical Management of Heart Failure within Elderly Population: the HEARTFAID Platform, Proc. of ITAB, Ioannina, Greece, 26-28 October 2006.
4. Conforti D. et al., HEARTFAID: A Knowledge Based Platform for Supporting the Clinical Management of Elderly Patients with Heart Failure, The Journal on Information Technology in Healthcare, 4(5), pp. 283-300, 2006.
5. Duftschmid G., Gall W., Eigenbauer E., Dorda W., Management of data from clinical trials using the ArchiMed system, Med. Inform. Internet, 27(2), pp. 85-98, 2002.
6. Gamberger D., Prcela M., Jović A., Šmuc T., Parati G., Valentini M., Kawecka-Jaszcz K., Kononowicz A. A., Candelieri A., Conforti D., Guido R., Medical Knowledge Representation Within Heartfaid Platform, Healthinf, Funchal, Madeira – Portugal, 2008.
7. Helms R. W., Entering Data from Remote Terminals in Clinical Centers using IBM's OS/TSO in the Kidney Transplant Histocompability Study, Technical Report 007, Chapel Hill, NC University of North Carolina, KTHS Statistics and Data Management Center, Department of Biostatistics, 1973.
8. Helms R. W., Data Quality Issues in Electronic Data Capture, Drug Information Journal, 35, pp. 827-837, 2001.
9. Johnson R., Hoeller J., Arendsen A., Risberg T., Sampaleanu C., Professional Java Development with the Spring Framework, John Wiley & Sons, 2005.
10. Merzweiler A., Weber R., Garde S., Haux R., Knaup-Gregori P., TERMTrial – terminology-based documentation systems for cooperative clinical trials, Comput. Meth. Programs Biomed., 78, pp. 11-24, 2005.
11. Michelsen L., Pedersen S. S., Tilma H. B., Andersen S. K., Comparing different approaches to two-level modelling of electronic health records., Stud. Health Technol. Inform., 116, pp. 113-118, 2005.
12. Minter D., Linwood J., Hibernate From Novice to Professional, Apress, 3 edition, 2006.
13. Nadkarni P., QAV: querying entity – attribute – value metadata in a biomedical database, Comput. Meth. Programs Biomed., 53, pp. 93-103, 1997.
14. Nadkarni P. et al., Organization of Heterogeneous Scientific Data Using the EAV/CR Representation, J. Am. Med. Inform. Assoc., 6(6), pp. 478-493, 1999.
15. Nadkarni P., Brandt C., Marenco L., WebEAV: Automatic Metada-driven Generation of Web Interfaces to Entity-Attribute-Value Databases, J. Am. Med. Inform. Assoc., 7(4), pp. 343-356, 2000.
16. Prcela M., Gamberger D., Bogunovic N., Developing Factual Knowledge from Medical Data by Composing Ontology Structures, MIPRO 2007, Opatija, Croatia.
17. Sonicki Z., Gamberger D., Smuc T., Sonicki D., Kern J., Data mining server: On-line knowledge induction tool, in: Proc. of Medical Informatics Europe, IOS press, pp. 330-334, 2002.
18. Wyatt J. C., When to Use Web-based Surveys, J. Am. Med. Inform. Assoc., 7(4), pp. 426-430, 2000.
19. HEARTFAID Consortium, D28 – Integration and Interoperability middleware prototype, 2008.
20. Altova XMLSpy, http://www.altova.com/xml-editor/
21. Form Handler, http://www.formhandler.net
22. HL7, Health Level 7, http://www.hl7.org
23. IHE, Integrating the Healthcare Enterprise, http://www.ihe.net
24. Instant Survey, http://www.instantsurvey.com
25. Mule, ESB http://www.mulesoft.org/display/COMMUNITY/Home
26. Survey Monkey, http://www.surveymonkey.com
27. Zoomerang, http://www.zoomerang.com

Appendix 1 – Knowledge model of the HEARTFAID eCRF, Simplified – Part 1 of 2

Anamnesis		
blood_pressure_change	(a,f)	enum[change]
bradycardia	(b)	boolean
bradycardia_change	(a,f)	enum[change]
chest_pain	(b)	boolean
chest_pain_change	(a,f)	enum[change]
chest_pain_remote	(b)	boolean
dyspnoea	(b)	boolean
dyspnoea_change	(a,f)	enum[change]
dyspnoea_remote	(b)	boolean
fatigue	(b)	boolean
and 19 more fields		

Echocardiography		
aorta_ascending_aorta_diameter	(b,a,f)	double
aorta_root_diameter	(b,a,f)	double
contractility_akinesis	(b,a,f)	boolean
left_atrium_anteroposterior_diameter	(b,a,f)	double
left_ventricle_end-diastolic_diameter	(b,a,f)	double
left_ventricle_end-diastolic_volume	(b,a,f)	integer
mitral_valve_deceleration_time	(b,a,f)	integer
mitral_valve_emax-amax	(b,a,f)	double
mitral_valve_mitral_regurgitation	(b,a,f)	integer
pulmonary_artery_pressure	(b,a,f)	integer
and 16 more fields		

24 h Holter Electrocardiography		
atria_ fibrillation_flutter	(b,a,f)	boolean
conduction_abnormalities	(b,a,f)	boolean
conduction_abnormalities_details	(b,a,f)	textfield
date	(b,a,f)	date
heart_rate_HF	(b,a,f)	double
heart_rate_LF	(b,a,f)	double
heart_rate_pNN50	(b,a,f)	double
heart_rate_rMSSD	(b,a,f)	double
heart_rate_SDANN	(b,a,f)	double
heart_rate_total_power	(b,a,f)	double
and 9 more fields		

Final Visit		
date	(f)	double
required_hospitalization_date	(f)	double

Rehabilitation		
model	(f)	textfield
required	(f)	boolean
time	(f)	integer

Chest X-ray		
cardio-thoracic_ratio	(b,a,f)	integer
comment	(b,a,f)	textarea
date	(b,a,f)	date
pulmonary_congestion_or_oedema	(b,a,f)	boolean

Quality of Life Questionnaire		
date	(b,f)	date
minnesota_total_score	(b,f)	integer
sf36_bodily_pain	(b,f)	integer
sf36_general_health	(b,f)	integer
sf36_mental_component_summary	(b,f)	integer
sf36_mental_health	(b,f)	integer
sf36_physical_component_summary	(b,f)	integer
sf36_role_emotional	(b,f)	integer
sf36_role_physical	(b,f)	integer
sf36_social_functioning	(b,f)	integer
and 2 more fields		

Family History		
primary_cardiomyopathy	(b)	boolean

Beat-to-beat Blood Pressure Monitoring		
baseline_finger_BP_SBP	(b,f)	integer
baseline_finger_HR	(b,f)	integer
comments	(b,f)	textarea
cuff_size	(b,f)	enum[cuff_size]
date	(b,f)	date
device	(b,f)	enum[device]
end_standing_CB_finger_BP_SBP	(b,f)	integer
end_standing_CB_finger_HR	(b,f)	integer
finger	(b,f)	enum[finger]
hand	(b,f)	enum[hand]
and 13 more fields		

Drug Theraphy		
drug_theraphy	(b,a,f)	drug
drug_theraphy_change	(a)	drug

Laboratory Assessment		
ALT	(b,a,f)	double
AST	(b,a,f)	double
blood_samples_for_DNA-RNA	(b,a,f)	boolean
BNP	(b,a,f)	pmol_mg
creatinine	(b,a,f)	umol_mg
creatinine_clearance	(b,a,f)	double
date	(b,a,f)	date
glucose	(b,a,f)	mmol_mg
glycated_hb	(b,a,f)	double
hb	(b,a,f)	double
and 12 more fields		

Physical Examination		
body_temperature	(b,a,f)	double
diastolic_blood_pressure	(a,f)	integer
heart_murmurs	(b,a,f)	boolean
heart_murmurs_apex	(b,a,f)	boolean
heart_murmurs_base	(b,a,f)	boolean
heart_murmurs_diastolic	(b,a,f)	boolean
heart_murmurs_systolic	(b,a,f)	boolean
heart_sounds	(a,f)	boolean
heart_sounds_bilateral	(b,a,f)	boolean
heart_sounds_fourth	(b,a,f)	boolean
and 24 more fields		

Cardiopulmonary Exercise Testing		
AT	(b,f)	double
BP_baseline_DBP	(b,f)	integer
BP_baseline_SBP	(b,f)	integer
BP_end_DBP	(b,f)	integer
BP_end_SBP	(b,f)	integer
BP_peak_ex_DBP	(b,f)	integer
BP_peak_ex_SBP	(b,f)	integer
data_recorded	(b,f)	boolean
O2_pulse	(b,f)	double
RQ	(b,f)	double
and 10 more fields		

Additional Visit		
date	(a)	date
next_scheduled_visit_date	(a)	date
other_than_chf_reasons_of_visit	(a)	boolean
required_advice	(a)	boolean
required_advice_details	(a)	boolean

Appendix 2 – Knowledge model of the HEARTFAID eCRF – Simplified – Part 2 of 2

Six-minute walking test		
BP_baseline_DBP	(a)	integer
BP_baseline_SBP	(a)	integer
BP_end_DBP	(a)	integer
BP_end_SBP	(a)	integer
date	(a)	date
HR_baseline	(a)	integer
HR_end	(a)	integer
SpO2_baseline	(a)	integer
walking_distance	(a)	integer

Lifestyle Information		
alcohol_use	(b)	boolean
physical_activity	(b)	enum[ph_activity]
smoking	(b)	boolean
smoking_cessation	(a,f)	boolean
smoking_cessation_date	(a,f)	date
smoking_duration	(b)	integer
smoking_no_cigarettes	(b)	integer

Non Cardiovascular Medical History		
anemia	(b)	boolean
anemia_worsening	(a,f)	boolean
bronchial_asthma	(b)	boolean
connective_tissue_diseases	(b)	boolean
diabetes	(b)	boolean
diabetes_type	(b)	enum[type12]
diseases_not related_to_hf	(b)	boolean
diseases_potentially related_to_hf	(b)	boolean
endocrine_disorders	(b)	boolean
exposure_to_endemic_diseasesy	(b)	boolean
and 29 more fields		

Demographic Data		
birthday	(b)	date
death	(a,f)	boolean
death_cause	(a,f)	textfield
death_date	(a,f)	date
sex	(b)	enum[sex]
status	(b)	enum[pat_status]

Cardiovascular Status		
aortic_regurgitation	(b)	boolean
aortic_stenosis	(b)	boolean
CABG	(b,a,f)	boolean
cardiovascular_reason_of_death	(a,f)	boolean
cerebrovascular_events	(b,a,f)	boolean
changes_in_therapy	(a)	boolean
chf_status_improved	(a)	boolean
chf_status_requires_hospitalization	(a)	boolean
congenital_heart_disease	(b)	boolean
congestive_heart_failure	(b)	boolean
and 41 more fields		

12-Lead Electrocardiography		
conduction_LBBB	(b,a,f)	boolean
conduction_PQ	(b,a,f)	integer
conduction_QRS	(b,a,f)	integer
conduction_QT	(b,a,f)	integer
conduction_RBBB	(b,a,f)	boolean
date	(b,a,f)	date
heart_rate	(b,a,f)	integer
heart_rate_24h_max	(b,a,f)	integer
heart_rate_24h_mean	(b,a,f)	integer
heart_rate_24h_min	(b,a,f)	integer
and 14 more fields		

Substudy 1 - Inclusion Criteria		
age_gt_65	(b)	boolean
chf	(b)	enum[diag_chf]
diastolic_dysfunction	(b)	boolean
ef_lt_40p	(b)	boolean
functional_capacity	(b)	boolean
hypertension	(b)	boolean
idcm	(b)	boolean
ihd	(b)	boolean
informed_consent	(b)	boolean
sinus_rhythm_presence	(b)	boolean
and 2 more fields		

Substudy 1 - Exclusion Criteria		
AIDS	(b)	boolean
autoimmune_disorders	(b)	boolean
cardiac_resynchronization_therapy	(b)	boolean
drug_or_alcohol_abuse	(b)	boolean
gfr_lt_30	(b)	boolean
hepatic_disease	(b)	boolean
immunosuppressive_therapy	(b)	boolean
malignancy	(b)	boolean
no_informed_consent	(b)	boolean
pacemaker	(b)	boolean
and 3 more fields		

Patient	
id	long
uuid	string
initials	string
usercenter	integer
createTime	date
updateTime	date
createUser	string
updateUser	string