An architecture for linking medical decision-support applications to clinical databases and its evaluation

Efrat German *, Akiva Leibowitz, Yuval Shahar

Medical Informatics Research Center, Department of Information Systems Engineering, Ben-Gurion University, P.O. Box 653, 84105 Beer-Sheva, Israel

**A R T I C L E   I N F O**

Article history:
Received 1 May 2007
Available online 7 November 2008

Keywords:
Medical decision-support systems
Databases
Knowledge bases
Medical records systems
Mapping

**A B S T R A C T**

We describe and evaluate a framework, the Medical Database Adaptor (MEIDA), for linking knowledge-based medical decision-support systems (MDSSs) to multiple clinical databases, using standard medical schemata and vocabularies. Our solution involves a set of tools for embedding standard terms and units within knowledge bases (KBs) of MDSSs; a set of methods and tools for mapping the local database (DB) schema and the terms and units relevant to the KB of the MDSS into standardized schema, terms and units, using three heuristics (choice of a vocabulary, choice of a key term, and choice of a measurement unit); and a set of tools which, at runtime, automatically map standard term queries originating from the KB, to queries formulated using the local DB’s schema, terms and units. The methodology was successfully evaluated by mapping three KBs to three DBs. Using a unit-domain matching heuristic reduced the number of term-mapping candidates by a mean of 71% even after other heuristics were used. Runtime access of 10,000 records required one second. We conclude that mapping MDSSs to different local clinical DBs, using the three-phase methodology and several term-mapping heuristics, is both feasible and efficient.

© 2009 Published by Elsevier Inc.

**1. Introduction**

The medical decision-support community suffers from a communication failure. The terms that one medical decision-support system (MDSS) uses to describe patient findings are often not recognized by another MDSS or local clinical database (DB), a problem pointed out already in the previous decade [1]. Not one MDSS can readily be used within all the different medical institutions. The reasons for this failure include the high variability of knowledge-based systems, the internal standards they employ, and the representational heterogeneity of clinical DBs. The importance of knowledge base (KB) reuse has been extensively studied [2–4]. Thus, the current study focuses on facilitating the reuse of decision-support applications and of the KBs associated with them, within institutions with heterogeneous clinical DBs, by overcoming the representational heterogeneity of the clinical DBs. The goal is not the integration of the various intra and inters institutional resources, but the production of a platform for applications that need to gather clinical data from various institutional DBs, and to apply heterogeneous medical knowledge to these data.

Institutions rarely share a common vocabulary of medical terms or the same encoding standards; thus, when either a KB or a DB is to be shared, a translation (or mapping) tool is required. In addition to vocabulary (terminology) differences, there are differences in the DB schema models and in the measurement units. For example, a local DB might store hemoglobin values in a table called “Hemoglobin Data”, refer to the hemoglobin concept as “HGB”, and use a non-standard unit to store its values.

Different MDSSs can potentially use the same KB or DB for different purposes, such as when one MDSS uses the KB and DB for monitoring patients, while the other MDSS can use the same KB and DB for runtime guideline application. One of the major motivations for our study was the desire to support both use and reuse of medical data and knowledge base.

The reuse of decision-support applications and of knowledge bases associated with these applications, across institutions with heterogeneous clinical DBs, is important in the following cases:

1. When several different MDSSs use the same domain KB, which needs to be applied to different clinical DBs. when an MDSS such as the IDAN [5] temporal-abstraction mediator, which supports queries about time-oriented clinical data or a runtime guideline application system such as the Spock system [6] needs to monitor and query patient data, refer to the same domain, e.g., oncology.
2. When the same MDSS applies several different domain KBs to the same clinical DBs (e.g., when one MDSS system (e.g., IDAN) uses KBs from both the oncology and cardiology domains to query the same clinical DB.)
As we shall show, our comprehensive solution involves the performance of three complementary tasks: first, we have developed a set of tools for embedding standard terms and units within KB's of MDSSs. Second, we have developed a set of methods and tools for mapping the local DB schema, terms and units into standardized schema, terms and units (e.g., the HL7 Reference Information Model (RIM), the LOINC observation codes and uniform unit's ontology) that are relevant to the KB of the MDSS. Third, a set of tools which, at runtime, automatically map queries formulated using the standard terms used by the KB of the MDSS, to queries formulated using the local schema, terms and units; thus, a single generic query from an MDSS can retrieve data from a variety of local clinical DBs. In order to keep these tasks complementary and independent we must separating the mapping of the database schema and the mapping of the terms process.

Clinical guidelines (GLs) are a powerful method for standardization and uniform improvement of the quality of medical care. When guidelines are used for decision-support within a patient's management, it is essential to use a mediator to connect to the patient's specific data which can be stored in specific EMR. MEIDA can be used as part of this mediator to support the challenge of bringing the gap between the GL knowledge base and the EMRs terms. Although the MEIDA system can be useful in mapping the raw terms of the leaves of the abstraction tree into standard terms, MEIDA does not build the abstraction tree. An abstraction of an abstract concept such as "bone marrow toxicity", of a time-oriented concept, such as "two weeks of moderate anemia", and even of more complex temporal patterns can be acquired from medical experts using a specialized temporal-abstraction knowledge-acquisition tool [7]. However, in this paper we will assume that a forest of such [temporal] abstraction trees, derived from raw clinical concepts, has already been defined in the knowledge base, and can be used by the MDSS, but needs to be mapped to a local clinical database.

In this research we have developed and implemented a framework, for linking knowledge-based MDSSs to multiple clinical databases, using standard medical schemata and vocabularies. Our research based on seven main research questions that examine the MEIDA three complementary tasks to bridge the gap between the MDSS knowledge base and the EMRs concepts. The methodology was successfully evaluated by mapping three KBs to three DBs. Using a unit-domain matching heuristic reduced the number of term-mapping candidates by a mean of 71% even after other heuristics were used.

1. A road map to the paper

In the following sections, the methods and components used in the MEIDA architecture are described in detail. In the Section 2) several previous studies concerning linking to MDSS heterogeneous DB's are presented. In Section 3, the MEIDA framework is introduced, its main components, users and implementation notes are presented in detail. Section 4, describes the evaluation of the MEIDA framework during the phases of the KB specification time, the DB set-up time, and data access runtime, using three different DBs; Section 5 describes the results. The last section contains the summary, discussion and conclusions.

2. Background

2.1. Methodologies for linking decision-support applications to heterogeneous databases

The largest barrier to linking knowledge-based MDSS's to heterogeneous DBs is the variety of ways in which similar data are represented in different DBs. This variety causes difficulty in sharing of decision-support applications across institutions having heterogeneous clinical DBs. A description of methodologies designed to overcome this barrier follows.

Health Level Seven is one of several American National Standards Institute (ANSI) standards, in the healthcare arena. HL7 provides standards for interoperability that improve care delivery, optimize workflow, reduce ambiguity and enhance knowledge transfer. At the core of the HL7 version 3 standards development methodology is the RIM [8], which is a static object-oriented model in UML notation. The RIM serves as the source from which all specialized HL7 version 3 information models are derived and from which all HL7 data ultimately receives its meaning. MEIDA used the RIM as a virtual-schema that enables interoperability between the MDSS knowledge base and the EMR structure.

Sujansky et al. [9–10] developed a prototype of the standard query model framework, called “TransFER”. The purpose of TransFER was to facilitate the sharing of decision-support applications across institutions with heterogeneous clinical DBs. The TransFER model provided a mechanism to customize DB queries automatically based on a reference schema of clinical data proposed by Sujansky. The TransFER methodology comprised a semantic data model, called FER (Functional Entity-Relationship model), for encoding site-independent clinical DB schema. This clinical DB schema can represent different relational DB schema. The schemas can vary in several respects, including: the identifiers used to denote certain entity types; whether data is stored or derived; and the representation of type hierarchies.

The Extended Relational Algebra (ERA) mapping language was another component of the TransFER methodology. A mapping between the standard queries models and specific relational DB implementation was defined by assigning an ERA expression to each construct that appears in the FER schema. For example, the entities type Patient and the relationship function Name-of was assigned the following schema-specific ERA expressions:

(Schema 1) Project [Name] (Patient)
(Schema 2) Project [PName] (SELECT [Type="PT"] (Person))

The purpose of both the TransFER methodology and our purpose is the same: to facilitate the sharing and reuse of decision-support applications across institutions with heterogeneous clinical DBs; however, the TransFER methodology refers only to the variety in the DB schema, and does not offer a solution to the variety in the terminology (term and unit). The TransFER methodology uses it own site-independent clinical DB schema, instead of using a schema based on an international medical standard.

Shaker et al. [11] focused on creating a data access system that provided bi-directional translation and mapping of data between heterogeneous DBs and a mediated schema. Semantic mapping rules stored in a KB were used by their generalized software to convert XML query results obtained from each data source to a common schema representing a single ontology. However, for each data source, a specific wrapper needed to be construction. One API requirement for a wrapper was that it produces a valid XML document which can readily be mapped from source to mediated schema using some set of reserve mapping rules.

There are several studies that focus on automated concept matching between laboratories DBs; Sun [12] used semantic network representations to model underlying native DBs and to serve as an interface for DB queries. His algorithm used vocabulary links to the Unified Medical Language System (UMLS) metathesaurus [13] to find matching nodes. Each semantic network node had an UMLS link that was represented by a list of metathesaurus concepts with semantics that were compatible with the node.
from the two semantic networks match if they have any common elements in their UMLS links, in his testing, the matching algorithms identified all equivalent concepts that were present in both DBs, and did not leave any equivalent concepts unmatched.

A popular standard vocabulary for lab terms is Logical Observation Identifiers Names and Codes (LOINC) [14]. Lau et al. [15] developed a method for the automated mapping of laboratory results to terms within the LOINC vocabulary. The first step of this method was to create sets of relationships for existing LOINC codes based on LOINC attributes. Similarly, laboratory results from a legacy system give its set of relationships according to the same LOINC attributes. Then, an automated comparison of the “to be matched” laboratory result’s relationship set to all the LOINC relationship sets, attribute to attribute is performed, which identifies the exact LOINC code match.

The methods of Sun and of Lau focused on automated mapping of laboratory results; in this study, in contrast, we focus on mapping the local clinical DB to a MDSS knowledge base, to enable the use and reuse of a decision-support application and of the KB associated with it, within institutions with heterogeneous clinical DBs.

Several studies [17,18] on sharing clinical guideline among institutions with different environments defined a theoretical approach based on a shared schema of the Electronic Medical Record (EMR). This shared schema was referred to as the virtual EMR. Mappings were created from the virtual EMR to actual EMR systems. The RIM was suggested as the basis for developing a standard interface to the EMR system.

The GuideLine Interchange Format (GLIF) is a model for representing guidelines in a machine-readable format [16]. GLIF was developed by the InterMed Collaboration project to enable the exchange of clinical practice guidelines among institutions and computer-based guideline applications. GLIF was designed to support computer-based execution by inclusion of a superset of Arden Syntax’s logic grammar as a formal expression language for specifying decision criteria and patient state. GLIF used a domain object model that enables GLIF steps to refer to patient data items, which are defined by a controlled terminology that includes standard medical vocabularies (e.g., UMLS) as well as standard models for the medical concepts (e.g., the HL7 Reference Information Model). However, in the final stage of creation of a guideline specification, an encoding team creates an implementable specification, in which data and action specifications are mapped onto specific data and procedures used by the implementing institution. The encoding team performs contextual adaptation of encoded guidelines for local health-care setting, and maps encoded guideline variables, concepts, and action specifications to the local clinical information systems. However, to the best of our knowledge, this approach was never implemented or evaluated.

A comprehensive research by Peleg et al. [19] introduced the computer-interpretable guidelines Knowledge-Data Ontological Mapper (KDOM) which enables bridging the gap from abstractions used in computer-interpretable guidelines (CIGs) to specific EMRs. Briding the gap involves: (1) using an ontology of mappings, and an optional Reference Information Model, to map an abstraction gradually into EMR codes, and (2) automatically creating SQL queries to retrieve the EMR data. Their research focuses on mapping abstractions to specific EMRs while MEIDA focuses on mapping the CIG terms to specific EMRs using medical standard vocabulary, and challenges the unit conversion problem. While MEIDA system can be useful in mapping the raw terms of the leaves of the abstraction tree into standard terms, MEIDA does not build the abstraction tree.

The openEHR endeavor deals with creating specifications, open source software and tools aimed at creating high-quality, re-usable clinical models of content and process, along with formal interfaces to terminology. The openEHR archetype framework is described in terms of Archetype Definitions and Principles [20] and an Archetype System [21]. Using the openEHR will enable future-proof information systems in a relatively simple information models and database schemas completely outside the software.

There are several commercial tools that connect heterogeneous DB schemata, not necessarily in the medical domain. Typical examples include the BizTalk suite of tools by Microsoft Co. [22], the dbMotion™ Solution package by dbMotion Co. [23], or the WebSphere Business Integration Connect by IBM International Co. [24]. BizTalk is based on a messaging component that provides the ability to communicate with a range of other software. It relies on pluggable adapters for different kinds of communication, using Web services and many others protocols. The dbMotion™ Solution creates secured Virtual Patient Records by logically connecting a group of care providers and organizations without data centralization. The dbMotion™ package enables healthcare organizations to securely share patient-centric medical information, regardless of the geographical and operational distribution of the data with all information remaining in its original format, location, system and ownership. The IBM WebSphere® Business Integration Connect Enterprise Edition is a fully functional, enterprise-strength Community Integration Gateway that helps meet the needs of companies looking to build operational B2B environments.

These tools, however, do not necessarily refer to the medical domain, do not have a solution for the variety inherent in medical vocabularies with respect to terms or units, and typically focus on integration of systems within the same institution (e.g., integrate several hospital centers of the same HMO), that is, on linking DBs to DBs, and not on linking KBS to DBs. To the best of our knowledge, no comprehensive, integrated set of tools that performs all of the three tasks discussed above (embedding standard terms within KBs of MDSSs; mapping the local DB schema, terms and units into standardized form; and providing runtime access to the local DB) was implemented and formally evaluated.

2.2. Overview of the medical database adaptor framework

Our generic methodology focuses on three modes (see Fig. 1): KB specification time, DB set-up time and data access runtime. At KB specification time, standard terms and units are embedded within a declarative (e.g., temporal pattern) or procedural (e.g., clinical guideline) knowledge base. In the KB specification phase MEIDA can be useful in mapping the raw terms of the leaves of the abstraction tree. However, MEIDA does not build the abstraction tree; that is specified using an external tool, such as Protégé or any other knowledge-acquisition tool (KAT). Regarding the DB set-up time, our methodology is based on using a virtual standardized view of the local clinical DB that we have defined and a set of standardized vocabularies for representing the raw-data terms in the KB used by the MDSS. We then map each local DB to (1) the MEIDA virtual medical record schema [25], (2) a set of standardized terminologies, and (3) a set of standard units. (For heuristic-strategy reasons, units are mapped before terms). These mapping allow the MDSS at runtime to formulate queries without knowing the complex and heterogeneous nature of the specific DBs. To be truly sharable, and avoid the “curly brackets” problem [26] when applying a MDSS in a new environment, the medical knowledge needs to be represented in a standardized form.

To perform the KB specification time, the DB set-up mappings and also support their data access at runtime application, we developed a framework called the Medical DB Adaptor (MEIDA) [27]. In the following sections, we describe the components of the MEIDA framework, and how we evaluated that framework.
3. Methods: The knowledge base—database link

The purpose of the MEIDA [27] framework is to facilitate both the use and the reuse of decision-support applications and of the KB’s associated with them, within institutions with heterogeneous clinical DB’s (or, alternatively, reuse a link to the same DB by a new MDSS application) by overcoming the representational heterogeneity of the clinical DBs. In order to achieve this purpose, the MEIDA framework focuses on three modes: KB specification time, DB set-up time and runtime data access. At KB specification time, standard terms and units are embedded within a declarative (e.g., temporal pattern) or procedural (e.g., clinical guideline) KB. In the KB specification phase MEIDA can be useful in mapping the raw-data level of the abstraction tree. At DB set-up time all of the mappings between the KB and the DB are performed to prepare the ground for accessing the data at runtime. At DB set-up time all of the mappings between the KB and the DB are performed to prepare the ground for accessing the data at runtime. Thus, at DB set-up (mapping) time, we need the help of the local DB administrators (DBAs), who use our tools to map their DB schema to the virtual-schema, as well as the help of a terminology expert, to map the local terms and measurement units to standard ones. At runtime data access, the MDSS can query the local DB using standard schema, terms and units from the KB, without knowing the schema and terminology of the local DB (see Fig. 2).

It is important to emphasize that the lack of a mapping of some part of the local schema into the virtual one (or any of the local terms and units into standard ones, as we will discuss in the following sections) implies that these particular data can not be accessed by any MDSS who is not intimately familiar with the local DB’s structure and terms.

3.1. The MEIDA user types and tasks

The users of the MEIDA application can be divided into two types:

- Users who help to map the local schema and terminology at KB specification and DB set-up time.
  - Local clinical domain expert during KB specification time
    Requires support for embedding standard terms and units within the KB.
  - Local DB administrators during DB set-up time
    Require support for the mapping of the local DB schema into the virtual medi cal record schema.
  - Local terminology experts during DB set-up time:
    Require support for mapping of local relevant (to the MDSS KB) application terms and units to standard vocabulary ones.

- Users who benefit from the system at runtime.
  - Runtime users of an MDSS
    Runtime users of an MDSS application may use the MEIDA search engine to understand the meaning of the application terms, or to find a term they want to be displayed regarding the patients, but whose local term they do not know. Mostly, however, they are indirectly using the MEIDA DB access module to access the local DB’s patient data, through the MDSS.

3.2. Use of the MEIDA system at knowledge base specification time

At KB specification time, the local clinical domain expert embeds standard terms and units within a declarative (e.g., temporal pattern) or procedural (e.g., clinical guideline) KB, using the MEIDA search engine service. The result is a KB grounded to standard terms and units that can be linked to any local clinical DB, after the performance of the DB set-up (mapping) time process. In the KB specification time MEIDA can be useful in mapping the raw terms of the leaves of the abstraction tree. However, MEIDA does not build the abstraction tree; that is specified using an external tool.
3.3. Use of the MEIDA Ontology–mapping modules during the database set-up time

3.3.1. Mapping the database schema

Since all clinical DBs represent, in principle, similar types of information, but each local clinical DB implements different data models and uses a different data schema, in this study a virtual medical record schema [25] was implemented. The virtual medical record schema is an essential part of MEIDA, and is based on the Reference Information Model (RIM) structure of the HL7 version 3 standard [8], and enables a uniform representation of any local clinical DB and frees the MDSS from the need to understand the details of DB schemas, by querying the local DB with a predefined known schema. The use of international terminology standards made the use of the same virtual-schema for all potential clinical DBs possible (see Fig. 3).

To reduce the complexity of each DB, the terms of the local DB were divided conceptually into four ontologies: Observations, Procedures, Medications and Diagnoses. For each type of ontology, a matching virtual medical record schema was developed. For example the virtual medical record schema for the observation ontology’s type may contain fields such as “Value”—the information that is assigned by the value of observation action, or “EffectiveTime”—the time at which the observation holds for the patient (valid time) or Text/Title for description of the observation by which a specific act may be known among people.

The overall schema (meta) ontology is based on the pre-existing HL7 medical standard, and the virtual-schema is defined so as to link the local database schema and the selected ontology type.

We provide support to the local DBA in mapping the DB schema to the virtual medical record schema. In the first step, the DBA selects the ontologies she wants to map, and then supplies the general parameters for her DB: the name of the server, DB, user and password. The system automatically displays the list of all the user’s tables in that DB. The DBA has the option of creating views that will serve her in the future mapping. After selecting the tables/views that need to be mapped, she maps each relevant field from her local tables to the virtual-schema field.

For example, in the case of the IDAN temporal-mediator architecture, which was used in the evaluation phase of this study, the main medium virtual-schema fields necessary for its operation include the following five core fields: patient ID, concept name, start time, end time and value. The reason is that most of the IDAN temporal-abstraction queries eventually mapped to observation, such as LOINC codes.
It is apparent from the literature that researchers have so far achieved little success mapping ontologies automatically; expert human input is essential in almost all cases [28]. In the application described here, we assisted the DBA by providing a suggestion for mapping the local DB fields to the virtual fields. For each field in the virtual medical record schema, we added a list of its synonyms; information-retrieval technology was used to search for a matching between the local table's fields and the virtual medical record schema fields and synonyms. We also checked that the type of the local fields match the type of the virtual fields.

3.3.1.1. Complex mappings. Not all the local data fields map in their existing form to the virtual fields. In several cases, it was necessary to convert a string to an integer, or map only part of the local DB field to the best matching virtual field. To achieve this, transformation functions were added that can be applied to the local DB fields before mapping them to the virtual fields.

The transformation functions included six categories: (1) String: includes all the main functions that can be applied on strings such as: Replace; (2) Field combination: enables the mapping of several local fields to one virtual field, such as when a date is built from the day, month and year. (3) Convert: enables the user to convert from one type to other, e.g., from string to integer; (4) Date manipulation: enables the user to add days or times to the original date field; (5) Expression: enables the user to develop an arithmetical expression using several fields; and (6) Set value: enables the user to set a fixed value to the virtual field, such as when we need to tag a field with the value ‘true’, to show that some action does take place, without actually needing the specific value of that action.

Using these transformation functions, new (derived) types of clinical information that are needed by the MDSS can also be created, such as the Body Mass Index (BMI) (Weight/Height²), which simplifies the query process, by treating such types as primitives instead of computing them in the MDSS.

3.3.1.2. The result of the schema mapping. The result of the schema mapping is an XML file, partitioned by the ontologies that have been selected by the DBA. For each ontology schema, the matching between the local DB fields and the virtual medical record schema fields, and any manipulation that needs to be done, is saved. From all these mappings an SQL string is automatically generated which will enable the MDSS to query the local DB without the need to know its schema. The basic format of the SQL string is:

```
Select localTable1.localField1 as virtualField1, localTable1.localField2 as virtualField2 FROM localTable1.
```

Although the main virtual-schema fields, such as patient ID, are assumed to exist in every local DB, figure automate mapping saves the local DBA the need to write SQL statement.

For example, the local DB might include a table named “labTestResult” that contains the patient’s lab-test results. This table contains four columns: patientID for the patient id, testName for the lab test name (e.g., WBC), testDate for the time that the test takes place and testValue for the value of the test lab result. These fields are mapped to the virtual medical record schema fields: id, displayName, date, and value, correspondently.

```
Select labTestResult. [patientID] as id, labTestResult. [testName] as displayName, labTestResult. [testDate] as date, labTestResult. [testValue] as value
From labTestResult
```

"From labTestResult"

Note that the above format allows also for transformations of the local field before being returned as the virtual field. The transformation (e.g., BMI = Weight/Height²) is defined within the generic query.

In this example, there is a table name BMIResult, with four columns: IDNum, date, Weight, Height. The displayName virtual field
receives the fixed value of ‘BMI’; the value fields receive the value of the result of the transformation.

Select BMresult.IDNum as id,

'BMI' as displayName,
[Weight]/ ([Height] [Height]) as value,
BMresult.[date] as [date]

From BMresult
Since it is not known which query will be entered at runtime, the final SQL statement is a union of all the individual ones. The MDSS uses this SQL sentence to query the local DB, thus avoiding the necessity of knowing the name of the tables and/or the columns of the local DB. At runtime the actual parameters that have been sent from the MDSS (i.e., patient id, concept) are added to this SQL sentence, as it’s “where” part.

Select labTestResult. [patientID] as id,
labTestResult. [testName] as displayName,
labTestResult. [testDate] as [date],
labTestResult. [testValue] as value

From labTestResult
Where id='123' and displayName='WBC'

3.3.2. Mapping the local database’s units
The next task is to map the local measurement units, which might be unknown, and might even be non-standard. We provide support to the local terminology expert in mapping the local unit. The starting point in the mapping is defining basic measurement domains (e.g., weight). The basic measurement domains are the most primitive domains. Basic measurement domain relations are the basic relations between two domains such as weight/volume. Every basic measurement domain has several basic measurement components (e.g., milligram); in this application, one basic measurement component can be translated into another basic measurement component in the same domain, with a predefined transformation. For example, the basic measurement domain weight can have the basic measurement components gram and milligram; the transformation from gram to milligrams is $\times 1000$.

We enable the local terminology expert to map any local measurement unit to a basic measurement component or basic measurement components relation. If no basic measurement component matches exactly, the local terminology can be enhanced by a transformation of a local measurement unit, so that it will match an existing basic measurement component. In some cases the basic measurement domain relations is a transformation that performs on the same domain; for example the local unit “m²” belongs to the basic measurement domain relation “Length x Length” and can be mapped to the basic measurement component relation “Meter x Meter”.

There is no common medical standard of units, but there are several measurement unit standards such as the one used by the LOINC vocabulary. In this application, LOINC's domain measurement units (e.g., Mass Concentration) were mapped to a basic measurement domain relation (e.g., Weight/Volume), and LOINC's units (e.g., G/L) were mapped to a basic measurement component relation. As we shall see, the unit mappings assist in the term-mapping task, by supporting various heuristics.

3.3.3. Mapping the local database’s medical terminology
To represent terms appearing inside various types of medical knowledge (e.g., expressions appearing within clinical guideline (GL’s)), a set of standard medical vocabularies was used, which enables users to share medical KBs that are not specific to any particular clinical DB, but that can be applied to each particular one. A vocabulary-server service was developed that serves as a search engine, which can be used by each local terminology expert to map the relevant local terms to standard medical terms (see Fig. 4). The current standard medical vocabularies include Logical Observation, Identifiers Names and Codes (LOINC) [15] version 2.04, Current Procedural Terminology (CPT) [29] version 4, International Classification of Diseases (ICD) [30] version 9 CM and the National Drug File (NDF) version 4.

One local term can be mapped to several standard medical terms; it is usually the case that the standard medical terms are more specific, so several specific standard terms can map to one generic local term. At runtime, if any of these standard terms appears in the MDSS query, it will map to the corresponding local term, without the need to be more precise in the query.

The purpose of the mappings is to increase the accessibility of the local DB and the sharability of the KB across institutions. We refer to the accessibility of a local DB as the ability to link the local DB to several different KBs; we refer to the sharability of the KB as the ability to link the MDSS’s KB to several different local DBs. There are three strategies for mapping local DB terms to MDSS’s KB.

1. Mapping all of the local terms: Mapping all of the local term to standard ones, using the vocabulary search service. After mapping all of the local terms, any medical decision-support system that uses only standard terms in its KB, can access and be applied to this local DB. This strategy increases the accessibility of the local DB; however it may reduce the sharability of the KB, since it might be the case that the local DB term(s) has been mapped to a standard term (or set of terms) different from the term(s) used by a particular MDSS KB; in that case, the MDSS might not be able to access the relevant local data without modifications to either the KB terms or to the DB mapping. Another disadvantage of this option is the time required by the local terminology expert to spend in the mapping process, which might involve potentially irrelevant local terms that no current MDSS refer to.

2. Incremental mapping: Mapping to standard terms only the local DB terms that need to link a specific MDSS’s KB. When a link between another MDSS and the local DB is needed, only the relevant terms that were not mapped in the previous phase need to be mapped. While mapping the relevant DB terms to the MDSS’s KB, validation of the fact that the mapping of the local DB terms to standard medical vocabularies are indeed the ones used in the MDSS KB is performed. This strategy increases the sharability of the KB; however it reduces the accessibility of the local DB, since a mapping is performed only to the local DB terms that need to be linked to a specific MDSS’s KB.

3. Mapping directly to the KB’s terms: In this option, the local terminology expert maps the local terms directly to the MDSS’s KB’s terms (all of which originated from a standard vocabulary, during the KB specification). This option might also be relevant when the KB’s terms are not necessarily represented in standard terms, although, of course, such non-standard representations will tend to reduce the sharability of the KB across institutions. This strategy limits the accessibility of the local DB to the specific MDSS.

The methodology for mapping local DB terms to a MDSS’s KB that was used in this study was a variation of the second strategy described above; in particular, it included the following steps:

1. Defining the medical domain for which the mapping will be performed (e.g., oncology).
2. Development of the MDSS’s KB that embeds standard terms and units.
3. Mapping to standard terms only the local DB terms that are relevant to the MDSS’s domain.
4. Validation that the mapping of the local DB terms to the standard medical vocabularies are indeed the ones used in the MDSS KB.
5. Iteration of steps 3 and 4 until no more terms need to be mapped.

Note, in this study only local terms needed by the KB were mapped.

3.3.4. Heuristics for mapping local terms

Our methodology for proposing standard vocabulary candidates to map from the local DB terms includes the use of three heuristics that can assist the local terminology expert in the terminology mapping process. The purpose of these heuristic is to cut down the number of irrelevant standard medical vocabularies candidate terms. The following three heuristics were used:

- Selection of the appropriate ontology—as a default, we search for a local DB term within the medical standard vocabulary (e.g., LOINC) that corresponds to the ontology that was selected for mapping of the database schema (e.g., Observations). The ontology heuristic cuts down considerably the number of potential candidates for mapping, typically by an order of magnitude.

- Full text search—search for candidates from the medical standard vocabulary that contain the name of the local term in their name or description fields. Microsoft full text search was used to search for candidate terms within the selected medical standard vocabularies. Matching the local DB term string typically reduces the number of candidates by an additional two orders of magnitude, as the evaluation has shown.

- Measurement unit heuristics—For local terms that have units, only standard terms whose units domain (e.g., Mass Concentration in the LOINC vocabulary) matches the local terms’ domain (e.g., Weight/Volume) are proposed for mapping. As will be shown in the section describing the evaluation, this heuristic leads to further significant reduction in the number of proposed candidates for terms to map into, often by another order of magnitude.

3.3.4.1. The MEIDA data access runtime module. At runtime, all the necessary mappings already exist as XML files; The MDSS gets queries from its users regarding clinical parameters or functions of clinical parameters. (Note: the query is often triggered due to an external input from the MDSS user, but might be completely internal to the process of applying the MDSS.) These queries are sent to the MEIDA Data Access Module (DAM) in order to query the local DB, and retrieve the result to the MDSS (using the Database Access Agent). We are assuming that the DAM is always a part of some mediator to the local clinical DB. (An example of a mediator, named IDAN, will be described later as part of the evaluation).

The DAM performs the following tasks (Fig. 5):

- Get the requesting MDSS queries: The DAM gets queries from the MDSS regarding clinical parameters or functions of clinical parameters (e.g., by asking about inferences deduced from these parameters).
- Authenticate runtime MDSS user: The DAM authenticates the MDSS runtime user’s authorization and refuses entry if the user is not authorized to access the specific data.
- Get local term and unit: The DAM translates the standard terms and measurement units to the local terms of the relevant clinical DB based on the mappings that were created at design-time.
- Get generic SQL statement: The DAM retrieves the local generic SQL that was created at design-time from the schema mapping file. This statement includes the union of all the relevant tables and fields of the local DB, which were mapped during the DB schema mapping process (for more detail see the “result of the schema mapping” at Section 3.3.1.2).
- Add constraints: The DAM adds to the SQL statement the relevant constraints (e.g., patient id, local concept) based on the query that was sent.
Query the local DB. The DAM queries the local DB with a translated query based on the local DB schema and terms and the constraints of the specific query parameters that were sent.

Perform a unit conversion: If the local DB uses units different from the standardized units, a conversion of the values is performed, for more detail see next section.

Encrypt the data: The DAM secures connectivity from remote servers (client DBs) through the Internet by encrypting the data over the Internet. The encryptions ensure that data remain private and confidential, and cannot be viewed by eavesdroppers who may be armed with network monitoring software.

Decrypt the resultant data set: A module of the DAM residing in the mediator's controller decrypts the resultant data set returned from the remote module of the DAM, to prepare it for display at the client browser.

Send the data to the requesting MDSS: The DAM sends the (possibly decrypted) corresponding dataset, expressed as a raw-data set in standard terms and units, to the requesting MDSS.

### 3.3.4.2. The unit's conversion

If the local DB uses units different from the standardized units, a conversion of the values is performed. In order to perform the conversion, not all candidate standard units must have matching local units; it is enough if one unit in each domain has a matching local unit, since the DAM can easily translate from one unit to other in the same domain with a predefined transformation function (see Fig. 6).

### 3.3.4.3. Implementation notes

The medical vocabularies are stored in an MSSQL server. The rest of the services were implemented in the Microsoft.Net environment, written in the C# programming language. The services interact with the MDSS using web-services (network services). All communication is performed using XML documents.

### 3.4. The evaluation methodology of the MEIDA architecture

To evaluate the MEIDA methodology and architecture, we have evaluated all of its three aspects, or phases: The KB specification time, The DB set-up time and the data access at runtime. Regarding the KB specification task we aimed to demonstrate the feasibility of using the MEIDA search engine service, to embed standard terms and units within a declarative (e.g., temporal pattern) or procedural (e.g., clinical guideline) KB for three different clinical applications. Regarding the DB set-up mapping process, we aimed to demonstrate the feasibility of using the MEIDA set of tools, to perform all of the relevant mappings, which are needed for the data access at runtime to the three respective clinical DB’s. Thus, only local terms needed by the KB were mapped. Regarding the runtime data access task we aimed to demonstrate the effectiveness of the actual data access process by measuring its performing performance aspects.

To evaluate the feasibility of these three tasks, we have used the MEIDA system to link the IDAN [5] temporal-abstraction mediation system. The goal of the IDAN framework is to answer complex temporal queries regarding both raw clinical data and its abstractions, such as are often required in medical applications. The temporal-abstraction knowledge base used by the IDAN system is assumed to include a forest composed of multiple derivation trees that derive in bottom-up fashion a complex concept in their root.
from a set of leaves of raw-data. Thus, for example, the concept of bone-marrow toxicity is abstracted in several stages from the raw-concept leaves platelet-count and white blood-cell count. Similarly, higher level temporal patterns are abstracted from their components, each of which might be an abstract concept itself. The IDAN temporal-abstraction database mediator is a modular approach designed for answering abstract, time-oriented queries. For example, “Did the patient have a period of more than 3 weeks of bone-marrow toxicity Grade 3 over the past 6 months?”

The IDAN mediator can be viewed either as an essential component of multiple medical DSSs that require access to the data stored within time-oriented clinical data sources, and need a knowledge-based interpretation of these data, or as a special type of a DSS that, focuses on the integration of multiple time-oriented data sources, domain-specific knowledge sources, and computation services. The mediator mediates abstract time-oriented queries from any application to the appropriate distributed components that can answer these queries.

However, the IDAN module requires actual low-level mapping and runtime access to link it to a local database’s data, while using the terms of the knowledge base that defines the relevant clinical temporal-abstractions (the knowledge is often provided by the medical application, such as a monitoring application, a visual-exploration module, a data mining system, or a guideline application architecture). These functions can be provided by the MEIDA architecture.

We have linked IDAN to three distinct clinical DBs: (1) an online retrospective DB of more than 1000 unidentifiable laboratory-test records of bone-marrow transplantation (BMT) patients at the Rush Medical Center, Chicago, USA. The Rush DB contains 2000 transactions per patient, thus a total of around 2 million transaction. (2) The BMT DB of the Hadassah Medical Center, Jerusalem, Israel, used for monitoring more than 2000 bone-marrow transplant patients throughout their course of illness. (3) A DB of patients monitored as being at high risk for coronary heart disease in the Soroka Medical Center, Beer Sheva, Israel, which includes a collection of 12,000 ambulatory and hospitalized patients, whose goal is aid primary physicians in properly identifying those patients requiring intervention or change of treatment. In each project, a copy of a limited amount of patients was used (those currently undergoing treatment). All patient identification was encoded and encrypted.

To evaluate the KB–DB mapping task, we have created, with the help of the respective medical domain experts, three different
declarative KB's in the three respective domains. The KB's defined temporal-abstractions useful for monitoring and therapy of patients in the respective domains.

To evaluate the DB set-up task, we worked with the local DBA of each DB, who provided us access permission and an explanation of the DB schema, and with a terminology expert who helped with the terminology mapping. Each of the local DB schemata was mapped to the MEIDA virtual medical record schema.

The DB terms that needed a link to the IDAN KB were mapped to standard medical terms using standard code from the LOINC, ICD-9, CPT and NDF standard vocabularies, using the MEIDA set of tools; the set of heuristics (describe in Section 3.3.4) were used to reduce the number of candidate terms for mapping. We performed a paired t-test in order to assess whether the reduction was significant.

To evaluate the performance of the runtime data access task, we have performed both a quantitative and a qualitative evaluation. In the case of the quantitative evaluation, we measured the time needed to send a query from the client browser to the DAM (Section 3.3.4.1), convert the standard terms and units to local ones, perform the query and return the result to the user. In addition, we have tested this process using a DB similar to the ones we used in this evaluation, using more than 10,000 patient data rows. We have tested every part of this process separately.

The qualitative evaluation included an evaluation of the overall temporal-abstraction (TA) mediator framework as part of the IDAN temporal-abstraction architecture, by linking the IDAN architecture to different DBs, using the MEIDA set of tools. Access to the clinical DB was implemented through the DAM module.

Our main research questions throughout the evaluation were:

1. What is the percentage of relevant local schema fields that are mapped to the virtual-schema fields successfully?
2. What is the percentage of the virtual-schema fields mappings that are suggested by the MEIDA system, which are approved by the DBA?
3. What is the percentage of local DB terms that are mapped into standard terms by the local terminology expert, using the MEIDA tools?
4. Does using one or more of our matching heuristics significantly reduce the number of candidate terms for mapping, (the "target set") compared to using none (or some of) the heuristics?
5. What percentage of the database terms that needed mapping required additional manual search by the KE (for example, due to the local term appearing in a manner different from known standards) after the use of all three heuristics? That is, what is the completeness of the overall mapping process?
6. What is the precision of the overall target-set filtering process, defined as the final percentage of relevant mapping candidates in the target-set, returned after using all three heuristics, namely, those which were actually mapped by the KE to the DB terms (i.e., the portion of relevant target-set terms, compared to the target-set gold-standard (GS))?

(7) What is the maximal time required to perform the term-mapping process for each DB?
(8) Can the IDAN mediator successfully and correctly access the three local DBs using the MEIDA runtime access system, within a reasonable time?

Note that the goal of the MEIDA system is to reduce the size of the target set of candidate terms from which the local terminology expert can select corresponding standard terms, to be mapped to the local DB; and similarly for the domain expert searching for a standard term to map to a particular KB raw term. Since the size of the potential target-set is huge (e.g. more than 300,000 terms in this study), it is infeasible to assess recall. However, it is indeed quite possible to assess precision for the overall target-set resulting from the application of all three heuristics, together, since the final target-set is typically small enough to be examined meticulously, and to judge exactly which terms were found useful by the KE for mapping into the local DB terms.

4. Results

We will describe the results of evaluating the MEIDA system within the IDAN temporal mediation.

(1) What is the percentage of relevant local schema fields that are mapped to the virtual-schema fields successfully?

Result:

Table 1 presents the results of mapping the DB schemata to the medical virtual medical record schema, while Table 2 presents the results of the automated mapping and Table 3 summarizes the use of the schema-field transformation function. Note that potential difficulties might occur when the mapping from the local DB schema to the RIM is not a 1:1 function. To counter that, we have used two options within the MEIDA system that were explained in the Section 3: First, any virtual field based on the RIM schema may be mapped to several local DB schema fields. For example, the virtual field "displayName", which describes the local concept name, may be found in several local tables and fields in the local DB schema. Second, when necessary, we can specify which transformation function (see Section 3.3.1) to use.

The schema mapping process in the case of the Soroka project was relatively the most complex; nevertheless, all (100%) of the relevant tables with their original schemata were mapped. During the mapping process, two views were created. The final generic SQL statement contains the union of 13 SQL statements. Each statement contains the result of one table/view fields mapping to the virtual-schema fields. The total time required in the case of the Soroka schema mapping process, using the MEIDA tools, was only 60 min. Most of the transformations that were required were due to the need to add fixed values to certain filed types (e.g., “Exists” to most events) and a type conversion (e.g., from String to Number).

The schema mapping process in the case of Hadassah project was the simplest: Since direct access to the Hadassah DB was not obtained, a copy of the data was used instead, which was already formulated closely to the virtual-schema of the IDAN application. No view needed to be created, and the final SQL statement

Table 1

Summary of the Database Schema Mapping process.

<table>
<thead>
<tr>
<th>Project</th>
<th>Num. of tables</th>
<th>Num. of views</th>
<th>Num. of SQL statements</th>
<th>Total time required</th>
<th>Mean of time required for mapping each local SQL statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rush</td>
<td>13</td>
<td>1</td>
<td>7</td>
<td>30 min</td>
<td>4.3</td>
</tr>
<tr>
<td>Hadassah</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>10 min</td>
<td>5</td>
</tr>
<tr>
<td>Soroka</td>
<td>21</td>
<td>2</td>
<td>13</td>
<td>60 min</td>
<td>4.6</td>
</tr>
<tr>
<td>Mean over all projects</td>
<td>21</td>
<td>2</td>
<td>13</td>
<td>33.3 min</td>
<td>4.5 min</td>
</tr>
</tbody>
</table>
Table 2
Distribution of the mapping fields that were suggested by the MEIDA system.

<table>
<thead>
<tr>
<th>Project</th>
<th>Total num of local fields for which a mapping was created</th>
<th>Num of mapping fields that were suggested by the MEIDA system (%)</th>
<th>Num of mapping fields that were approved by the DBA (%)</th>
<th>Num of mapping fields that were not approved by the DBA (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rush</td>
<td>35</td>
<td>15 (42.8%)</td>
<td>9 (60%)</td>
<td>6 (40%)</td>
</tr>
<tr>
<td>Hadassah</td>
<td>10</td>
<td>5 (50%)</td>
<td>3 (60%)</td>
<td>2 (40%)</td>
</tr>
<tr>
<td>Soroka</td>
<td>65</td>
<td>20 (30.7)</td>
<td>13 (65)</td>
<td>7 (35%)</td>
</tr>
<tr>
<td>Mean over all projects</td>
<td>17 (36.3%)</td>
<td>10.25 (62.5%)</td>
<td>6 (37.5%)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3
A summary of the use of the transformation function in the case of the three test applications.

<table>
<thead>
<tr>
<th>Project</th>
<th>String function</th>
<th>Date function</th>
<th>Convert function</th>
<th>Expression function</th>
<th>Add static value</th>
<th>Total use of transformation function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rush</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Hadassah</td>
<td>1</td>
<td></td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Soroka</td>
<td>6</td>
<td>1</td>
<td></td>
<td>11</td>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

- String function: includes all the main functions that can be applied on strings such as: Replace.
- Date function: enables the user to add days or times to the original date field.
- Convert function: enables the user to convert from one type to other, e.g., from string to integer.
- Expression function: enables the user to develop an arithmetical expression using several fields.
- Add static value: enables the user to set a fixed value to the virtual field.

Table 4
The number of terms at each medical standard vocabulary.

<table>
<thead>
<tr>
<th>Medical standard</th>
<th>Num. of terms</th>
<th>Projects in which the standard was used</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDF—4</td>
<td>286,257</td>
<td></td>
</tr>
<tr>
<td>Loinc—2.04</td>
<td>30,598</td>
<td>Rush, Hadassah, Soroka</td>
</tr>
<tr>
<td>ICD—9 CM</td>
<td>15,434</td>
<td></td>
</tr>
<tr>
<td>CPT—4</td>
<td>8,160</td>
<td>Rush</td>
</tr>
<tr>
<td>Total</td>
<td>340,449</td>
<td></td>
</tr>
</tbody>
</table>

Table 5
Summary of terminology mappings performed for the three local databases.

<table>
<thead>
<tr>
<th>Project</th>
<th>Num. of terms in local DB</th>
<th>Num of local DB terms for which a mapping was created</th>
<th>Num. of mappings performed into standard terms</th>
<th>Num of local terms for which no matching standard terms were found</th>
<th>Num of local terms for which an additional (manual) search was needed (%)</th>
<th>Mean num. of mappings created for each local term</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rush</td>
<td>594</td>
<td>154</td>
<td>154</td>
<td>0</td>
<td>60 (38.9%)</td>
<td>1</td>
</tr>
<tr>
<td>Hadassah</td>
<td>107</td>
<td>81</td>
<td>206</td>
<td>2 (2.5%)</td>
<td>30 (37%)</td>
<td>2.5</td>
</tr>
<tr>
<td>Soroka</td>
<td>1588</td>
<td>27</td>
<td>66</td>
<td>1 (3.7%)</td>
<td>6 (22%)</td>
<td>2.4</td>
</tr>
<tr>
<td>Mean over all projects</td>
<td>1588</td>
<td>27</td>
<td>66</td>
<td>0.7 (1.15%)</td>
<td>6 (22%)</td>
<td>1.6</td>
</tr>
</tbody>
</table>

Table 6 presents the result of the terminology mapping process. For only 1.15% of the local terms no matching standard term was found. Thus, for 98.85% of the local DB terms, one or more matching standard terms were found. In the case of the Rush project, for 100% of the terms a matching standard was found, since the Rush project mappings used both the LOINC and CPT standard (see Table 4), using the iterative mapping process described above. Most the terms for which no standard terms were found were events such as day of birth and date of death (these appear in the CPT vocabulary but not in the LOINC vocabulary, and we did not use their combination in all the projects).

(4) Does using one or more of our matching heuristics significantly reduce the number of candidate terms for mapping, (the "target set") compared to using none (or some of) the heuristics?

Result:
Table 6 presents the results of using the various heuristics to cut down on the number of candidates proposed by the MEIDA vocabulary search engine. At the beginning, the candidates could in theory be selected from any of our vocabularies (total of 340,449 terms). Using our first heuristic, the local terminology expert selects which medical standard to work with for each local term, which corresponds to the ontology that was selected (e.g., Observations or Procedures), thus reducing the number of proposed candidate terms to only those from the vocabulary actually chosen to be used in their application of each local term (e.g., a mean of 30,598 candidate vocabulary terms per local term when LOINC medical standard was selected). After using the “Full Text Search heuristic”, the number of candidate terms for mapping reduces dramatically again (e.g., to a mean of 72 ± 109 across all sites, weighted by the
number of local term searches). Note that in the case of Rush, either CPT or the LOINC vocabularies or both were used to map the local terms; in the case of Hadassah and Soroka, only LOINC happened to be relevant for the particular applications to be used and for the local terms that were considered to be relevant.

In the case of the Soroka project, in which a complete DB with measurement units was available, the unit-domain matching heuristic was used. Using the heuristic reduced the candidate terms lists on average by an additional 71% after using the basic free-text search heuristic; that is, to only 29% (on average) of the terms returned by the ontology-specific free-text similarity search (from a mean of 99.26 ± 137 proposed candidates to a mean of 28.89 ± 43.57 proposed candidates). The maximum improvement was 97.7%, the minimal improvement was 22.22%, and the median of the improvement was 74.6%. Note that these results refer to 27 separate searches (see Table 5). When we performed a paired T-test, we found that the improvement of the unit-domain matching heuristic was indeed highly significant (p = 0.005).

(5) What percentage of the database terms that needed mapping required additional manual search by the KE (for example, due to the local term appearing in a manner different from known standards) after the use of all three heuristics? That is, what is the completeness of the overall mapping process?

Result:
The results in Table 5 show that in most cases (98.5%), at least one mapping to a standard vocabulary was in fact performed for each of the terms from the three local databases. For only 1.5% of the local terms no matching standard term was found and necessitated additional manual search by the KE.

(6) What is the precision of the overall target-set filtering process, defined as the final percentage of relevant mapping candidates in the target-set, returned after using all three heuristics, namely, those which were actually mapped by the KE to the DB terms (i.e., the portion of relevant target-set terms, compared to the target-set gold-standard (GS))?

Result:
After using all three heuristics the mean number of candidates terms proposed by the search engine was 28, from this target size the local terminology expert selected an average of 2.3 terms for each local term. Thus the precision of the overall target-set filtering process was 2.3/28 = 0.08.

(7) What is the maximal time required to perform the term-mapping process for each DB?

Result:
Table 7 presents the average time required to perform the term-mapping process for each project. It can be seen that the longest time was required in the case of the Hadassah project, where the largest number of mappings was performed. Experience with the mapping tools and the search engine service on the part of a medical knowledge engineer demonstrated a significant shortening of the time needed for the mapping, and has reduced the average time needed to map a single term from 10 min to 3 min during the project [31].

(8) Can the IDAN mediator successfully and correctly access the three local DBs using the MEIDA runtime access system, within a reasonable time?

Result:
We managed to access from the IDAN system all three DBs successfully. We have assessed the MEIDA runtime aspect and found it as completely functional, in the sense that it supported all data access queries to all three DB’s during runtime. The results have shown that it took less then 1 s to accessed the local DB and retrieve the relevant data. Furthermore, querying a DB similar to the ones we used in this evaluation, but using 10,000 rows, took 7 s; most of the time was spent on fetching the data from the server to the client browser within our intranet, while the actual query computation time (mapping and accessing) required around one second.

5. Summary, discussion and conclusions

5.1. Summary of the study

In this study, we focused on facilitating the use and reuse of decision-support applications and of the KBs associated with them, within institutions with heterogeneous clinical DBs, by overcoming the representational heterogeneity of the clinical DBs. We have described a complete framework, the MEIDA architecture, for linking clinical DBs to KBs of MDSSs.

The MEIDA framework focuses on three modes: KB specification time, DB set-up time and runtime data access. At KB specification time, standard terms and units are embedded within a declarative (e.g., temporal pattern) or procedural (e.g., clinical guideline) knowledge base. At DB set-up (mapping) time, all the DB mappings are performed; thus, we need the help of the local DB administrators (DBAs), who use our tools to map their DB schema to the virtual-schema, as well as the help of a terminology expert, to map the local terms and measurement units to standard ones. At runtime data access, the MDSS can query the local DB using standard schema, terms and units from the KB, without knowing the schema and terminology of the local DB.
During the DB set-up time, MEIDA supports three main tasks: mapping the local DB schema to a virtual medical record schema that enables the representation of any local clinical DB, mapping the local terminology to a standard based on a set of predefined medical vocabularies and mapping the local measurement units into standard ones. We developed tools that at runtime automatically map queries formulated using standard terms as are used by the KB of a medical decision-support system, to queries formulated using the local terms and schema. Thus, at runtime, the MEIDA system receives from the MDSS a request for patient data. It uses the KB–DB mappings that have been performed at DB set-up time, creates the appropriate SQL statement to query the local clinical DB, retrieves the corresponding result, and reports it back to the MDSS.

Using this architecture, any medical decision-support application can automatically access patient data by bridging the gap between the local DB’s data representation and the virtual-schema and terminology assumed by the MDSS’s knowledge base.

In the evaluation, we used the MEIDA system to link the IDAN temporal-abstraction mediation system to different clinical DB’s. We linked the IDAN architecture to three different clinical DBs from three different clinical centers. The results show that most (98.8%) of the terms in the local database that were considered as potentially relevant to one of the two applications we had in mind were mapped to standard vocabulary terms. During the process of schema mapping, 62.5%, on average, of the suggestions for virtual-schema mappings (see Table 2) were deemed as relevant (suggestions were made by matching local names to the name or list of synonyms, of the virtual fields).

Indeed, when mapping the local terminology to a standard one, the local terminology expert was considerably supported in her search for candidate standard terms by the three heuristics. The selection of the appropriate ontology reduces the number of candidates by an order of magnitude: the “search” heuristic, which uses the vocabulary-server’s search engine, reduced the number of candidate terms by another two orders of magnitude, to a weighted mean of 72 ± 109 candidates out of 30,598 terms, in the case of the LOINC vocabulary (see Table 4). Note that the standard deviation was very high, indicating that each search was quite independent from other searches; the range between the minimal value and maximal value was very high. These two heuristics are well known, although our study strived to verify their benefit formally. However, rather surprisingly, using the unit-domain heuristic further reduced the candidate terms list, in the case of the Soroka project, by another order of magnitude—by a mean reduction of an additional 71% (see Table 6) and sometimes up to 97%, leading to a reduction by a mean factor of 3.4 in the number of proposed candidates.

Considering the first two heuristics from an information-retrieval measures point of view, both are potentially prone to loss of recall (due to potential loss of relevant mapping candidates) and thus, potentially, even to loss of precision (due to possible reducing the proportion of relevant candidates). It is indeed difficult to rule out such cases of loss of additional relevant candidates (for the same reason that it is difficult to estimate recall and even precision in the case of searching the World Wide Web, due to its sheer size). Indeed, there was, for example, a small number of potentially relevant candidates within the CPT or ICD vocabularies, in addition to the LOINC vocabulary, that might have been missed due to the very ordering the first two heuristics (that is, we only applied the string/synonym search within the vocabularies suggested by their term-mapping tool user as part of the first, ontology selection, heuristic). Furthermore, it should be remembered that in most cases (98.5%), at least one mapping to a standard vocabulary was in fact performed for each of the terms from the three local databases, proving that at the very least, the use of the various heuristics did not seem lead to loss of all potential mapping candidates. Furthermore, the unit-domain heuristic is monotonic in its level of recall, in the sense that, per definition of the rigid unit-domain constraints, no candidates eligible for mapping are lost by application of these constraints; and it also enhances significantly (presumably, by the same rate of reduction that we had witnessed, namely, about 3.4:1) the level of precision (proportion of relevant candidates) within the list of final proposed terms. Indeed, in the case of the Soroka DB, the mean number of eventually relevant standard vocabulary candidates for mapping each local term stayed constant, and thus, the proportion of relevant mapping terms actually increased from a rate of 1.6% (1.6/99.26) after using the first two heuristics (ontology selection, and the free-text synonym search) to a rate of 5.5% (1.6/28.89) after adding the unit-domain heuristic.

We consider three heuristics to be sufficient since using more than three heuristics is likely to lend to a plateau in the precision.

With respect to performance, at runtime, the process of sending a query from the client browser to the DAM, converting the standard terms to local one, creating the local SQL statement, performing the query and retrieving it to the user took less then 1 s. This was true both for a DB with 1000 patients, with a mean of 2000 transactions and for a 20,000 records simulation DB.

5.2. Discussion

Overall, the MEIDA 3-phase methodology and architecture were shown to be feasible and useful. As explained in Section 3, one of the goals of the MEIDA system during the DB set-up phase, is to reduce the size of the target-set of candidate terms from which the local terminology expert (or KE) can select corresponding standard terms, to be mapped to the local DB; and similarly for the domain expert searching for a standard term to map to a KB raw term. Since the size of the initial potential target-set is huge, it was infeasible to assess recall. However, as shown above, we found it quite possible to assess precision for the final target-set resulting from the application of all three heuristics.

It is important to emphasize that, MEIDA is not used to define the abstraction tree. That step was typically performed previously, within a dedicated knowledge acquisition tool, typically when defining the temporal-abstractions at the root of the abstraction tree. Usually the mapping was performed between the raw terms and the standard vocabularies. Sometimes we can map an abstract term directly to a corresponding standard vocabulary term, as in the case of Anemia.

Limitations of the approach:

We did not find any tool or process to compare with, that perform the overall task set of the MEIDA system, namely, both at design-time and at runtime, with respect to both mapping the knowledge and accessing the data. We did not want to compare MEIDA to itself, or to a part of itself (e.g., to a simplified version using no heuristics, which we felt would be a rather artificial control group).

However, a part of the design-time functions performed by the MEIDA architecture does exist in other systems as well. In particular, the vocabulary search tool used by MEIDA, when used only for general concept-based search (i.e., without reference to a particular database, in which the MEIDA system uses database-specific heuristics, such as the units) resembles several of the UMLS tools, which enable searching for concepts, based on keywords and on the source vocabularies.

A thorough study might be conducted to compare these specific functionalities and respective mechanisms in the future. For example, in our results we conclude that a proper assessment of recall cannot be performed, but that only for 1.15% of the local terms no matching standard term was found. A thorough comparison with the UMLS search mechanism may in the future provide
the ability to compare this particular function of the MEIDA system to a gold-standard, and better reveal whether this is indeed the case.

In the schema DB mapping, the implicit control group here is the explicit use of SQL statements. We acknowledge that at least in some cases, depending on how much the DBA is aquatinted with the local DB, much of the overall virtual-schema core-fields mapping effort might be invested in understanding the semantics of the local DB. Nevertheless, in this study we examined the precision of the automated suggestion, which we considered as useful in any reasonable scenario.

In this study we used a set of transformation function between units in the same domain. All of the transformations encountered for the mappings encountered in this particular study were supported; however, theoretically, we may in the future meet transformations that are not supported, such as transformations that use the log function.

It should be emphasized that only terms in the local DB that were mapped to some term in a standard vocabulary can be accessed; thus, random, ad-hoc queries using arbitrary vocabulary terms other than those to which a local term was mapped might not be able to access those terms. Our approach is thus specific to the mapping of the local terms to a set of predefined KBs and/or applications that the MEIDA mapping tools user has in mind.

The process of mapping local DB terms to standard vocabulary terms guarantees semantic consistency and leads to semantic disambiguation of potential conflicts among terms with similar names in different local DBs but different precise semantics. Of course, that very same semantic precision might lead, in certain cases, as pointed out above, to a potential situation in which there is no clear 1:1 mapping between a standard term used in the KB, and the standard term into which the “real” corresponding local term was mapped. That is, in theory, they may have been mapped to different terms. However, that very fact (which can be detected automatically) since there will be “dangling pointers” in both cases signals a need for adding another mapping, or, rarely, for modifying the KB.

The use of heuristics reduced the number of candidate terms for mapping in the target-set significantly. However, it was probably at the cost of decreased recall, that is, there might have been additional good mapping candidates that were omitted at an earlier phase. The fact that for approximately 1% of the local terms no mappings were found is potentially points to these false negative terms, which might have been relevant for mapping these local terms but were discarded by the filtering process. However, the third heuristic (matching the units domain) should not reduce recall, if our basic assumption regarding the existence of units in standard vocabularies and local DBS holds.

Additional advantages of the approach:

The very nature of the MEIDA architecture helps to protect the security and privacy of the local DB, since the MDSS does not need to know the schema of the local DB, nor the names of the local tables and their various fields. Essentially what are being used here are wrappers that encapsulate the access to the local DB, and expose just the virtual-schema.

As long as the MDSS KB is defined (at KB specification time) using only standard terms and units, our approach guarantees scalability, since mapping into (or, rather, from) a new local DB requires approximately the same time and effort regardless of the local DB size.

The MEIDA methodology enhances the accessibility of the local DB as well as the sharability of the MDSS’s KB. The accessibility of the local DB was defined as the ability to link the local DB to several different KBs, while the sharability of the KB was defined as the ability to link the MDSS’s KB to seven different local DBs.

5.3. Future work

The MEIDA framework can be useful in the KB acquisition and in the runtime application of clinical guidelines. The acquisition task enables conversion of free-text guidelines to a machine-comprehensible format. It can use the MEIDA vocabulary search engine service for generic specification of key terms, thus enabling reuse of the formal specification at multiple sites. Each term in the guidelines can be specified using a standard medical term to support use and future sharing and reuse. The clinical guideline application system can use the MEIDA runtime data access component for queries the local clinical DB, typically as part of a [temporal] abstraction mediator such as the IDAN architecture.

We have already started to assess this option with encouraging results, using the DeGel guideline library architecture. We have found it feasible to embed the MEIDA system within the DeGel guideline library architecture and to map most guideline terms, during guideline specification time, to standard terms. The mappings of the guideline terms should be able to support the reuse of the formal guideline specifications at multiple local clinical sites.

5.4. Conclusions

We conclude that mapping the terms and units of a medical knowledge base to different local clinical DBs, during design-time, using the three-phase methodology, including the several term-mapping heuristics, and applying these mappings at runtime to efficiently access the data, is both feasible and quite efficient.

After all of the heuristics were applied, from 22% to 38% of the local DB terms that required mapping still necessitated additional manual search. Thus, the methodology is not appropriate for performance of a fully automated mapping. A further underlying assumption of the methodology is that all of the local DB terms can indeed be mapped to standard ones.

The use of the three KB–DB mapping heuristics reduces dramatically (by about four orders of magnitude) the number of terms proposed to the knowledge engineer or local DBA as potential candidates for mapping. In particular, the use of the unit-domain heuristic reduced the number of candidate terms by a mean of a further 71%, even after the first two heuristics, ontology selection and synonym search, were used.

In summary, the methodology has been shown to be feasible and usable for both design-time and runtime KB to DB term-mapping.

Acknowledgments

This research was supported in part by NIH award no. LM-06806 and by Israeli Ministry of Defense Award BGU-No.89357628-01. We want to thank the staff of Ben-Gurion University’s medical-informatics laboratory for their useful comments. We also want to thank all of our collages at Stanford University and at the Palo Alto Veterans Administration Health Care Center, for their assistance in evaluating the IDAN architecture.

References


