A Model for Computing Temporal Eligibility Criteria on Large and Diverse Data Repositories

Adel Taweel, Elyes Lamine, Richard Bache

A Model for Computing Temporal Eligibility Criteria on Large and Diverse Data Repositories

Adel Taweel
Department of Computer Science
Birzeit University
Birzeit, Palestine
ataweel@birzeit.edu

Elyes Lamine
Toulouse University, ISIS,INU
Champollion, Castres, France
Toulouse University, IMT Mines Albi,
CGI, Albi, France
Elyes.lamine@univ-jfc.fr

Richard Bache
Department of Primary Care and PHS
King's College London
London, UK
richard.bache@kcl.ac.uk

***Abstract*—There have been numerous attempts to build *query generators* that compute eligibility criteria (EC) for a clinical trial automatically on repositories of patient data. However, one of the challenging key features of EC is the ability to express and compute complex temporal aspects. Existing EC generators has limited temporal capability and those do rely on underlying database technology to perform temporal reasoning. We propose a model that incorporates temporal features of existing generators. However, it separates the computation of the criteria, and in particular the temporal semantics, from the extraction of clinical data from the database to increase the efficiency of execution. We explain the implementation of this model and in particular its temporal algorithm, which runs in *O(n log(n))* time where *n* is the number of clinical facts stored making it more efficient than existing reported generators, where performance, at best, has been reported to be $O(n^2)$. We perform an empirical validation to demonstrate the results.**

Keywords— Eligibility Criteria, Query Generator, Clinical Trial, Temporal Reasoning

I. INTRODUCTION

Clinical sites such as GP practices and hospitals routinely store clinical data about patients, increasingly in a structured (as opposed to free text) format in electronic health record (EHR) systems. Clinical trials need to recruit patients according to a set of eligibility criteria (EC), which will often relate to the clinical facts stored in EHR systems. Naturally, there is growing interest in automating the identification of patients for clinical trials using EHR systems. To enable computerised identification of suitable patients first requires a formal and machine-readable expression of a set of EC for the trial in question. Where eligibility criteria express temporal conditions, as Ross et al. [1] have identified, in 40% of cases a means of formalising time and performing temporal reasoning is also required. Most existing approaches, in computing temporal queries, depend on using the query language of the underlying database or use combined query to reason over complex temporal ECs. We argue here that by formalising the expression of EC independently of the representation of the clinical data allows the temporal reasoning to be performed separately from the retrieval of the clinical data. This offers substantial improvements in the time taken to compute EC on large repositories of data. We described such a system and perform an empirical study to show that it does offer significant performance advantage over previously implemented systems.

There have been various attempts to formalise ECs with the ultimate aim of computing them automatically against data from EHR systems. This problem has been approached from two directions: formalising natural language ECs in general and computing a subset of computable ECs from actual clinical data. In the first, approaches such as EligWriter [2], ERGO [3], Gello [4] and Arden Syntax [5] seek to represent EC in a formal or semi-formal notation. These enable varying degrees of formal reasoning that is not possible with natural language representations. The EliXR [6] tool seeks to analyse natural language text to create a formal representation. However, although such notations are useful for communicating EC in an unambiguous way, they are currently not sufficient to compute patients' eligibility from repositories of clinical facts. This is partly due to the fact that they are not restricted to information that is likely to be recorded by EHR systems in a structured format. Furthermore, just because a concept can be formalised and rendered into a machine-readable notation does not necessarily mean that there exists an algorithm for computing it against structured data.

This paper defines an approach for expressing and computing EC from databases of clinical data with both relative and absolute temporal semantics comprising three components:

1. A data-source independent model of EC based on the capabilities of existing query generators.
2. A mechanism for extracting the relevant data from the data source and rendering it into a form that can be subsequently processed against complex temporal criteria.
3. An algorithm that takes the modelled EC and extracted data and calculates the set of eligible patients in *O(n log(n))* time where *n* is the number of clinical facts.

Our motivation for constructing this model was to develop a platform that enables a non-technical user to create a query at a remote workbench and then distribute this query to a number of remote sites to be computed over diverse patient data repositories and receive counts of eligible patients. For reasons of governance, any processing requiring access to sensitive patient data should be performed locally within the clinical site firewall so that only counts are returned to the researcher working outside the institution. It will also require secure communication technologies to link the user interface operated remotely by the researcher to components of the system lying within the clinical site's firewall. Although these communication technologies are beyond the scope of this paper, we do address the issue of expressing the query in a serialised form that these technologies would require. A key requirement is to ensure that computation of ECs is done efficiently. Thus, the approach was evaluated against a set of databases of varying sizes to determine its time efficiency.

The rest of the paper is organised as follows, section II reviews related work, section III describes the proposed

approach, section IV presents evaluation results, sections V and VI discuss and conclude the paper.

II. Related Work

Researchers have also pragmatically constructed working systems, which we shall term *query generators* that compute only a subset of all possible EC, specifically those that may be computed from the clinical data available. Such systems often provide a graphical user interface (GUI) e.g., FARSITE [7], VISAGE [8], PatternFinder [9], i2b2 [10], STRIDE [11], ePCRN [12], TRANFoRm [13, 25, 26] and Trial DB [14]. However, the DXtractor [15] system is purely textual. Two further technologies are also relevant here: Chronus II [16] and CLEF [17, 23, 24, 27]. Chronus II is an extension to SQL to allow temporal queries but is proposed for use in encoding EC. CLEF is a programming interface that allows the construction of queries in Java code. Table 1 shows a summary of the key features of each system. The ECs supported by these generators and related technologies are constrained in two ways:

- They must relate to information that is actually captured in the EHR systems in a structured format.
- Any criterion must be capable of being computed against that information without human intervention.

We note that these technologies may also be used to answer research questions other than how many patients are eligible for a trial.

TABLE I. Key Features of Query Generators Reviewed

Name	Technology Type	Temporal semantics
STRIDE	Graphical query generator	absolute time, age at event
FARSITE	Graphical query generator	time interval between actual date and event date
(Extended) TrialDB	Graphical query generator with text output	Subset of Allen's operators
PatternFinder	Graphical query generator	Allen's operators
i2b2	Graphical query generator	same visit as
Visage	Graphical query generator	same visit as, absolute time
e-PCRN	Graphical query generator	last, absolute time
DXtractor	Textual query generator	first, last, before, after, equals
CLEF	Java Libraries	before or after 3 anchors: birth, prior event or now, first, last
Chronus II	Query Language	Allen's operators

We argue that the ultimate aim of such work is to construct a system for automatically computing EC against patient data in general. However, there is an important limitation to this aspiration. Natural language criteria as expressed in clinical trials are often subjective, require some degree of clinical judgment or require information not generally held in patient records, at least as structured data. At best, for automatic EC computation to work with complete accuracy, it can only apply objective criteria to structured patient data. Therefore, there is a theoretical limitation to which EC can be computed. Nevertheless, such a system is still useful in that it can automatically create a shortlist of patients to whom the incomputable criteria can be applied manually.

The most general framework for temporal reasoning is provided by Allen's interval algebra [18]. This assumes that any clinical fact to be represented as an event associated with a (contiguous) time interval. Allen's algebra provides 13 different relations, which include *before*, *meets*, *overlaps*, *starts*, *finishing*, *equals* and the inverse of these relations. Arguably this provides far more expressive power than would be necessary for the EC routinely drafted by researchers or for the clinical data that would typically be found in EHR systems. The implementation of temporal semantics in the query generators cited above varies considerably, as shown in TABLE I. We distinguish here between *absolute* and *relative* temporal semantics. Absolute temporal semantics relate the date of a patient event, such as diagnoses, procedures, medications and lab test results to a specified time point, which may be an explicit date or implicitly the time at which the query submitted. Relative temporal semantics relate two or more events that apply to a specific patient such as a particular procedure performed at most one year after the first diagnosis of a (related) condition.

FARSITE, i2b2, VISAGE and ePCRN express only limited temporal semantics. In FARSITE only the time interval of a clinical event compared to the date on which the query is executed can be specified. ePCRN also supports only absolute time defined against the current time. Since the XML-based repository is a snapshot of a particular patient's details, it will contain only the most recent measurements of vital signs and lab tests. In i2b2 and VISAGE the only temporal constraint is that two clinical facts apply to the same encounter/visit. This is a limited example of relative temporal semantics since events in different encounters cannot be related.

In CLEF, constraints for clinical characteristics are defined first (any, first, most recent, all). Temporal relations can then be defined via temporal anchors. Two types of anchors are supported, an age-anchor and an event anchor. The age-anchor specifies a clinical event as basis for the age calculation, e.g., age at the time of diagnosis. Clinical characteristics may be defined relative to an anchor event described by the time interval, e.g., start of treatment with drug Y within a month from diagnosis. It should be noted that the implementation of the temporal semantics results in issues of time efficiency. As the authors concede "*performance was a problem*" [17].

In PatternFinder temporal queries are constructed via sentinel events (index event). The sentinel event is linked to an event in the past (baseline) and/or in the future (follow-on) using the relations: after, before, within x prior, within Y following and equal. A user interface to define sentinel, baseline and follow-on events and temporal relations has been implemented. Not clear whether this approach is restricted to link a maximum of three events via temporal relations or linked as a sub query in another query (e.g. Boolean link to other characteristics). However, Lam [9] does evaluate its time efficiency and point out that when using a

temporal condition of one event after (or before) another, the query can take hours to process. TRANSFoRm supports a similar approach, although no results of its query generator have been reported.

Trial DB (or strictly speaking an extension to Trial DB) specifies temporal relations via temporal operators linking two clinical characteristics. The temporal operators cover a subset of Allen's relations, including *before*, *after* and *during*. In addition, events can be characterized by start, end and duration. To support users, a graphical tool allows users to select the appropriate Allen operator when expressing temporal conditions. The restriction to temporal relations between two characteristics is a limitation, which may inhibit the intuitive creation of more complicated queries.

In STRIDE, filtering with temporal constraints can be performed for pairs of events. In a first step, operations are performed on the time-stamps of an event to select from three options: any, earliest, most recent (similar to CLEF). In the next step two events are set in relative order with three options: *follows*, *precedes* or *precedes or follows*. Then the range for comparison sets the time between two events and has three options: less than, greater than and between two values. For each criterion an age-anchor can be set. Again, the restriction to temporal relations between two characteristics is a limitation similar to Trial DB.

The DXtractor approach uses rules to define sets of patients satisfying each rule and then imposes further temporal constraints on each set of patients. Boolean and temporal operations can be performed by defining new sets operating on the existing sets. Similar to other approaches the earliest or latest event can be specified in case an event has several instances. Nigrin and Kohane [15] argue convincingly that by modelling all clinical facts as zero-duration events gives sufficient richness to express the temporal queries that would be required in the clinical domain. Where events are given a zero-duration time stamp, the 13 Allen operators collapse to just five: *before and after* (with optionally some specified interval of time), *equals* (simultaneous with), *first* and *last*. This approach is both elegant and generic since it is not restricted to relations between pairs of events (STRIDE) or triples of events (PatternFinder). However, no recent implementation of this approach could be identified.

Chronus II is a superset of SQL that allows temporal constraints using a 'WHEN' clause. Thus, it is not restricted to clinical data but was designed with this purpose in mind. Chronus II adopts a valid-time temporal model and supports states (interval time-stamps), events (instant time-stamps) and non-temporal tables. It implements all of Allen's 13 relations. Chronus II adapts TSQL2 [19] temporal query language to extend the standard relational model and the SQL query language to support temporal queries that include temporal projection, joins, granularity conversion and coalescing. It was not possible to derive the status of implementation and practical use of Chronus II from the literature or indeed its time efficiency.

Lam [9] identifies the fundamental problem with implementation of relative temporal semantics. If applied to a database of clinical facts it requires joins, and often self joins, across very large tables. Yet the problem of computing time is not inherently $O(n^2)$ but the problem of time efficiency arises when attempting to compute the criteria using a general-purpose query language such as SQL. To this end we propose the creation of a data-source independent Eligibility Criteria Model (ECM) that separates the computation of the ECs with temporal semantics from the extraction of data from the data source.

III. Proposed Eligibility Model

The Eligibility Criteria Model (ECM) was designed to be used on two different warehouses of clinical data where, except for demographic data (age, gender and death), each row represents an event with a simple time stamp. The two data warehouses were constructed with distinct schemata: the i2b2 schema [10] and a purpose-built native schema based on the HL7 Reference Information Model (RIM) [20]. However, we note the wide applicability of the single-timestamp approach since Deshpande et al. [14] and Nigrin and Kohane [15] observe that most clinical data is stored in this way. Furthermore, where an event has time stored as an interval (a two-timestamp approach) it can be represented as two one-timestamp events, namely the beginning and end of the activity.

A. Definition of Non-temporal Capabilities

The non-temporal capabilities of the query generators reviewed above all combine a set of *rules* where each rule applies to one patient attribute. These rules are linked with Boolean operators, logical operators, such as conjunction (AND) or disjunction (OR) to construct a set of EC. Each rule addresses a particular patient attribute e.g., age, gender, HBA1c results or diagnosis of a particular cancer. Therefore, each rule would be computed against a particular type of clinical fact e.g., dates of birth, recorded gender, lab tests and diagnoses respectively. Although natural language EC may make reference to several patient attributes in one criterion e.g., "*Male patients over 18 with lung cancer*," this would count as three rules for gender, age and diagnosis respectively. The terms '*atomic queries'* [1] and '*simple criterion'* [15] are also used in the literature as synonyms for *rule*.

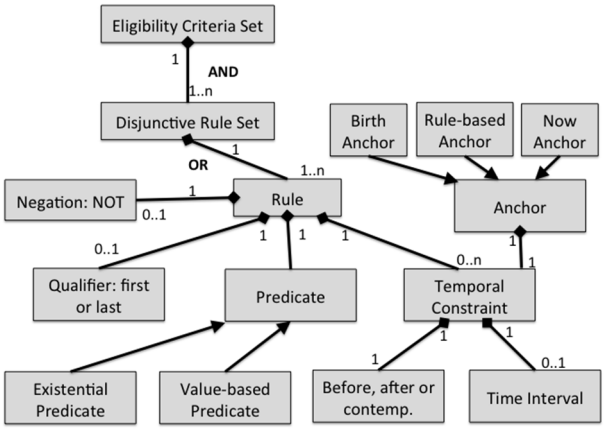

Fig. 1. ECM-Generic Temporal Model

The proposed ECM supports three types of predicate types: existential, numeric and categorical. There are certain ordinal scales such as the ECOG scale of ambulatory status that may express its value in either a numerical or coded value. For these data elements both categorical and numerical predicates were permitted.

B. Definition of Temporal Capabilities

The ECM assumes zero-duration events and each rule specified whether the event used in the predicate is the first or

last for that patient except for certain unique events such as birth where this qualifier is not needed. Any rule may have at most one temporal constraint relating the event in the predicate to one of three anchors. The ECM supports three temporal anchors: *now* (the time of the query), a *rule-based* anchor using the time of the event used in a previous rule and a *birth* anchor using the patient's date of birth. This means that age is not explicitly queried but expressed as a temporal constraint on the birth (being x years before the now anchor). Gender is determined by a gender identification event assumed to occur at birth. Thus, the ECM treats all clinical facts as zero-duration events making no distinction between demographic and other data. Fig. 1 shows the high-level object representation of the ECM

C. Interoperability

The ECM was defined to be independent of both the structure (database schema) and semantic representation (clinical coding systems used). Furthermore, where a rule applies to physical quantity and a unit of measurement is specified, it does not require that the repository use the same unit as the query. This is particularly important for lab tests when measuring the concentration of some substance in a urine or blood sample (e.g., liver function tests) where either mass or molarity per unit volume are routinely used.

The ECM contains an evaluation algorithm, which operates independently of the syntactic and semantic representation of the data repository. This approach requires a mechanism to extract the clinical facts needed to compute a particular rule and render them into a standard representation. Thus, to connect with each data repository, a set of adaptors was constructed [21] for each different repository type. This setup is shown in Fig. 2. The advantage of this approach is that the adaptors are lightweight when compared to the ECM and its evaluation algorithm and can rapidly be assembled for new types of repositories.

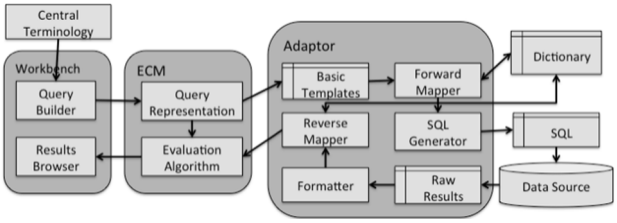

Fig. 2. System Architecture incorporating ECM

D. Eligibility Criteria Evaluation Algorithm

Each rule has a predicate that refers to clinical facts of some type e.g., a diagnosis for type II diabetes or administration of the drug metformin. The evaluation algorithm has three distinct steps:

1. For each rule we determine the set of patients that meet this criterion alone ignoring for the moment any NOT operators. This is achieved by running through each patient in turn and using the clinical facts related to that rule to determine if the patient is included in the set. Note that an absence of facts about a patient leads to exclusion.

2. We compute the Boolean operators linking the rules, AND, OR and NOT, by applying the respective operations intersection, union and set subtraction on the subsets of patients yielded by each rule. This is how the NOT operator is able to convert an inclusion criterion into an exclusion criterion.

3. Where counts rather than a list of patients are required, we determine the cardinality of the resultant subset.

In the first step, patients are computed for inclusion in the resultant set using the following steps:

a. The facts related to a single patient for a given rule are identified. So, for a rule including all patients with most recent BMI reading within 3 months is greater than 23, all BMI readings for that patient are considered

b. The temporal constraint is used to filter the facts, removing those that fall outside the specified time window. If there is no temporal constraint no facts are filtered out. In the example, all BMI readings more than 3 months old are excluded.

c. The first or last fact is selected, according to what is specified in the rule. In our example we select the last.

d. This single fact is used to compute the truth-value of the predicate to determine if the patient is in the resultant subset or not. The actual BMI reading is compared to the value 23. If it exceeds this then predicate is true and the patient is included. If the value is less than or equal to 23 or no reading exists then the predicate is false.

Note that for existential predicates, the mere presence of the fact makes the predicate true.

E. ECLECTIC Notation

A human-readable notation called ECLECTIC (Eligibility Criteria Language for European Clinical Trial Investigation and Construction) [22] was devised to give users of the system a clear and unambiguous description of the query. A set of ECs, which together form the query, is expressed as a list of consecutively numbered rules. Each rule defines the clinical event used to express the predicate. This may be existential or else value-based in which case there is an '*in*' clause to specify the reference range. Except for demographic events, such as *born*, *gender* and *deceased*, which are, by definition, unique events, there is also a *first* or *last* qualifier prefixing each event. Each rule may have a temporal constraint in which one of the temporal operators *before*, *after* and *contemp* is followed by one of the three anchors: *now*, *birth* or a *rule* specified by its number. The notation is intended to be readable by clinicians and is logged when any user queries a particular data repository, for governance reasons. Fig. 3 shows an example.

```
1 gender() in {[SNOMED Clinical Terms:248153007,"Male"]} and
2 first diagnosis([ICD-10:E11,"Non-insulin-dependent diabetes mellitus"]) and
3 first diagnosis([ICD-10:I50,"Heart failure"]) at least 5 year after rule(2) and
4 last vitalsign([LOINC:4548-4,"Hemoglobin A1c/Hemoglobin.total:Mass Fraction:Point in time:Whole
   blood:Quantitative"]) in range(>=7.0) unit([ucum:%,"percent"])
   at least 3 month before now
```

Fig. 3. Example of ECLECTIC

IV. Evaluation and Results

The model was validated by showing that an implementation was possible that was independent of either the GUI and the repository used to hold clinical facts. Such an implementation was shown to be compatible with a GUI created subsequently and was also shown to be able to access

diverse clinical data repositories. Benchmarking was performed by running a query with temporal constraints on a series of databases of increasing size to show that the observed relationship of time with database size is *O(n log(n)).*

A. Implementation of EHR4CR Platform

An implementation of the ECM has been constructed as part of the EHR4CR platform for computing eligibility criteria. This platform enables a user to compose a set of eligibility criteria at the *workbench* and then launch a federated query to obtain patient counts from nominated clinical sites. The query is accepted by the *orchestrator* and then sent on the nominated clinical sites, in this case hospitals. Each hospital contains an *endpoint*, which executes the query on the respective data warehouse that has been populated from the hospitals' EHR systems. It derives counts, which are sent back to the workbench, from which the query originated, via the orchestrator. The ECM represent both within the workbench and the endpoint. In the workbench it is used to express the composed query. It is then serialised and sent to the endpoint where it is de-serialised. Here the query is used to generate the SQL queries and the results are processed by the evaluation algorithm. Under construction is a variant that enables users at a single site to produce a list of eligible patients.

The EHR4CR currently uses two database designs: one developed in the project (the native database) and the i2b2 database (without the i2b2 query generator). Four i2b2 data warehouses and three data warehouses using the native database schema were populated using an ETL (extract-transform-load) process from hospital EHR (electronic healthcare systems). Multiple local terminologies were used. They have been queried by researchers working at eight Pharma sites.

A query expressed in the implementation of the ECM is a hierarchy of Java objects. This can be readily serialised and communicated via the Internet to remote data sources. Since the query is de-serialised within the firewall of the clinical site the evaluation algorithm is executed within the firewall. Thus for feasibility only these counts are returned to the workbench and the user at the workbench has no direct access to sensitive clinical data.

B. Benchmarking of Evaluation Algorithm

To demonstrate that the evaluation algorithm was (at worst) *O(n log(n)),* a query with temporal constraints was run on a series of databases each an order of magnitude larger than the last. Times for the extraction of data from the database and the evaluation of the algorithm were measured to show that these times did not increase by more than an order of magnitude. Real clinical data from 100 diabetic patients was obtained and this was used to populate a database. This data was then replicated 10, 100, times etc. A query was constructed so that it would make use of all three types of temporal anchor and would also yield a non-zero number of eligible patients.

Fig. 4 shows the ECLECTIC for the benchmarking. The first rule uses a now anchor to include patients over 50 years old. The second uses a birth anchor to include patients who were first diagnosed with type II diabetes after their fortieth birthday. The third rule tests on the most recent BMI reading. The fourth rule tests on a BMI reading at least 6 months prior to the event used in rule 3. The experiment was run on a standalone version of the software. Instead of using the workbench, the query was hard coded. No remote communication across the Internet was used. It should be noted that the size of the query and results of counts would not be affected by the size of the database used. The databases were held in Microsoft SQL Server running on a virtual server with 4 virtual CPUs and GB of RAM, on a Zen Server. Database tables were indexed to improve efficiency. The SQL queries were launched from a Mac Book with a 2.4 Ghz processor and 4GB of RAM, which was also used to perform the evaluation algorithm. The two were connected by a local network. For each database the query was run 10 times and the mean time was calculated.

```
1 born() at least 50 year before now and
2 first diagnosis([ICD-10:E11,"Diabetes mellitus type 2"]) at least 40 year after
        birthdate and
3 last vitalsign([SNOMED Clinical Terms:60621009,"Body mass index"]) in range(<=30.0)
      unit([UCUM:kg/m2,"KiloGramsPerSquareMeter"]) at most 1 year before now and
4 last vitalsign([SNOMED Clinical Terms:60621009,"Body mass index"]) in range(<30.0)
      unit([UCUM:kg/m2,"KiloGramsPerSquareMeter"]) at least 6 month before rule(3)
```

Fig. 4. ECELECTIC Used for the Benchmarking

TABLE II. shows the number of seconds taken to perform the database queries and also to perform the evaluation algorithm as well as the total time for both. The factor increases in the times as the database increase with size is also shown. Fig. 5 represents the data in graphical form. To be consistent with an *O(n log(n))* algorithm, the factor increase in time for an order of magnitude increase in the database size should, at worst, barely exceed 10. In reality it never exceeded 10. Since the experiment shows an increase of 4 orders of magnitude, any part of the algorithm that was $O(n^2)$ or higher would manifest itself with factor increases greater than 10.

TABLE II. RESULTS OF BENCHMARKING

Database Size (Patients)	Database Size (Clinical Facts)	Running DB Queries (sec)	Factor increase	Evaluation Algorithm ECM-(sec)	Factor increase	Combined	Factor increase
100	8.8K	0.4	-	1.8	-	2.2	-
1K	88K	0.7	1.8	2.3	1.3	3.0	1.4
10K	880K	2.3	3.3	3.5	1.5	5.8	1.9
100K	8.8M	8.8	3.8	11.7	3.3	20.5	3.5
1M	88M	82.7	9.4	103.2	8.8	185.9	9.1

V. DISCUSSION

The only query generator that mentions the algorithmic complexity of temporal reasoning explicitly is PatternFinder [9], which is $O(n^2)$ time for one event before or after another event. Profiling of PatternFinder showed that queries could take hours to perform. This is due to a self-join on a (typically large) database table holding clinical facts. It is reasonable to assume that if other implementations, addressed the temporal semantics at the SQL level, they would also suffer a similar problem. To express four cycles of chemotherapy, that is four events each after the other, the complexity becomes $O(n^4)$ making the approach not scalable to typical data warehouses holding $>10^8$ clinical facts. The algorithm used in the ECM avoids this problem by transferring the problem to an application programming language, where unlike SQL - a declarative language, there is explicit control of the algorithm.

VI. CONCLUSION

The ECM expresses queries in the form of a set of ECs with temporal constraints and supports both absolute and relative temporal semantics. The ECM and its associated

implementation are able to support federated queries from a workbench to multiple clinical sites with diverse data warehouses differing in both structural and semantic representation. This is made possible by separating the query model and its evaluation algorithm from the data sources holding the clinical data. The use of a Java object model means that the query may be serialised to enable communication between the workbench used by the researcher and the clinical sites to be queried. The evaluation algorithm has been shown to run in *O(n log(n))* time and thus addresses the performance issues that have beset other attempts in implementing temporal semantics.

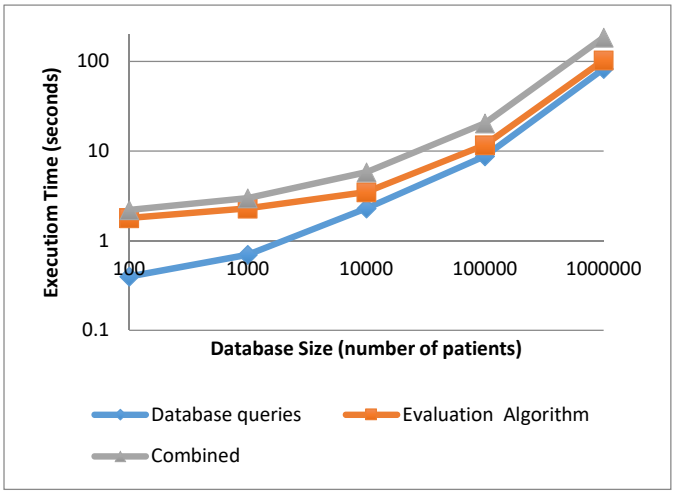

Fig. 5. Execution Time Against Database size (number of patients)

ACKNOWLEDGMENT

This research is supported by the National Institute for Health Research (NIHR) Biomedical Research Centre at Guy's and St Thomas' NHS Foundation Trust and King's College London.This work was part funded by the IMI-funded collaborative project EHR4CR (Electronic Healthcare Records for Clinical Research), Grant agreement no.: 115189. We would like to acknowledge the work on designing and implementing the EHR4CR platform performed by many members of the project.

REFERENCES

[1] Ross J, Tu S, Carini S, and Sim I. Analysis of Eligibility Criteria Complexity in Clinical Trials. AMIA Summits Transl Sci Proc. 2010; 2010: 46–50

[2] Gennari J, Sklar D, Silva J. Cross-tool communication: From protocol authoring to eligibility determination. In: Proc AMIA Symp; 2001. p. 199-203.

[3] Tu S, Peleg M, Carini S, Bobak M, Rubin D and Sim I, A Practical Method for Transforming Free-Text Eligibility Criteria into Computable Criteria

http://www.bioontology.org/sites/default/files/a%20practical%20method%20for%20transforming.pdf- last accessed 9/7/2021

[4] Sordo M, Boxwala A, Ogunyemi O and Greenes R. Description and status update on GELLO: a proposed standardized object-oriented expression language for clinical decision support. Stud. Health Technol. Inform. 2004;107:164–8.

[5] Health Level Seven International, Arden Syntax, available at http://www.hl7.org/special/Committees/arden/ - last accessed 9/7/2014

[6] Weng, C, Wu X, Luo Z, Boland MR, Theodoratos D, and Johnson SB, EliXR: an approach to eligibility criteria extraction and representation, J Am Med Inform Assoc. vol. 18, suppl. 1, 116-24, 2011

[7] Ainsworth J, Buchan I. Preserving consent-for-consent with feasibility-assessment and recruitment in clinical studies: FARSITE architecture. *Studies in Health Information Technology. 2009;* 147: 137-148

[8] Zhang GQ, Siegler T, Saxman P, Sandberg N, Mueller R, et al. VISAGE : A query interface clinical research. AMIA Summits Transl Sci Proc. 2010. Mar 1; 2010:76-80.

[9] Lam S. PatternFinder in Microsoft Amalga: Temporal query formulation and result visualization in action. 2008; http://www.cs.umd.edu/hcil/patternFinderInAmalga/PatternFinderS-HonorsPaper.pdf - last accessed 9/7/2021

[10] Murphy SN, Weber G, Mendis M, Gainer V, Chueh HC, et al. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). J Am Med Inform Assoc. 2010; 17: 124-130

[11] Lowe HJ, Ferris TA, Hernandez PM and Weber SC. STRIDE- An integrated standards-based translational research informatics platform. AMIA 2009 Symposium Proceeding. 391-395

[12] ePCRN, The electronic patient care research network, http://www.epcrn. bham.ac.uk/ last accessed 9/7/2021.

[13] L. Zhao, S.N. Lim Choi Keung, J. Rossiter, T.N. Arvanitis, Query Formulation Workbench, Deliverable 5.3 available at http://transformproject.eu/Deliverables_files/D5.3%20Query%20Formulation%20Workbench.pdf, last accessed 9/7/2021.

[14] Deshpande AM, Brandt C, Nadkarni PM. Temporal query of attribute-value patient data: utilizing the constraints of clinical studies. Int J Med Inform. 2003; 70: 59-77

[15] Nigrin DJ, Kohane IS. Temporal expressiveness in querying a time-stamp-based clinical database. JAMIA. 2000; 7: 152-163

[16] O'Connor MJ, Tu SW, Musen MA. The Chronus II temporal database mediator. Proc AMIA Symp. 2002; 567–571.

[17] Austin T, Kalra D, Tapuria A, Lea N and Ingram D. Implementation of a query interface for a generic record server. Int J Med Inform. 2008; 77: 754-764

[18] Allen JF. Maintaining knowledge about temporal interval. In: Communications of the ACM. 26 November 1983. ACM Press. pp 832-843

[19] Richard T. Snodgrass, editor, The TSQL2 Temporal Query Language, Kluwer Academic Publishers, 1995.

[19] Benson T, Principles of Health Interoperability HL7 and SNOMED: Chapter 7, Springer 2009.

[20] Bache R, Miles S and Taweel A, An Adaptable Architecture for Patient Cohort Identification from Diverse Data Sources, J Am Med Inform Assoc, 2013 doi:10.1136/amiajnl-2013-001858

[21] Bache, R., Taweel, A., Miles, S. and Delaney, B.C., 2015. An Eligibility Criteria Query Language for Heterogeneous Data Warehouses. *Methods of information in medicine*, *54*(01), pp.41-44.

[22] Chen, Yuhui, Richard Bache, Simon Miles, Marc Cuggia, Iñaki Soto-Rey, and Adel Taweel. An SOA-based Platform for Automating Clinical Trial Feasibility Study. In *Proceedings of the IADIS International conference e-Health 2013*, pp. 84-94. IADIS, 2013.

[23] Taweel, A., A. L. Rector, J. Rogers, D. Ingram, D. Kalra, R. Gaizauskas, M. Hepple et al., CLEF-joining up healthcare with clinical and post-genomic research, BJHC Books Ltd, 2004.

[24] Taweel, A., Rector, A.L., Rogers, J., Ingram, D., Kalra, D., Gaizauskas, R., Hepple, M., Milan, J., Power, R., Scott, D. and Singleton, P., CLEF-joining up healthcare with clinical and post-genomic research. BJHC Books Ltd, 2004.

[25] Ethier, J.F., Dameron, O., Curcin, V., McGilchrist, M.M., Verheij, R.A., Arvanitis, T.N., Taweel, A., Delaney, B.C. and Burgun, A.,. A unified structural/terminological interoperability framework based on LexEVS: application to TRANSFoRm. *Journal of the American Medical Informatics Association*, 2013, *20*(5), pp.986-994.

[26] Zhao, L., Lim Choi Keung, S.N., Taweel, A., Tyler, E., Ogunsina, I., Rossiter, J., Delaney, B.C., Peterson, K.A., Hobbs, F.D. and Arvanitis, T.N.,. A loosely coupled framework for terminology controlled distributed EHR search for patient cohort identification in clinical research. In *Quality of Life through Quality of Information,* 2012, (pp. 519-523). IOS Press.

[27] Rector, A.L., Rogers, J. and Taweel, A., Models and inference methods for clinical systems: A principled approach. In *MEDINFO 2004* (pp. 79-83). IOS Press.

[28] De Lusignan, S., Krause, P., Michalakidis, G., Vicente, M.T., Thompson, S., McGilchrist, M., Sullivan, F., van Royen, P., Agreus, L., Desombre, T. and Taweel, A., 2012. Business process modelling is an essential part of a requirements analysis. *Yearbook of Medical Informatics*, *21*(01), pp.34-43