Cross-Trial Query System for Cancer Clinical Trials

Radu Calinescu, Steve Harris, Jeremy Gibbons and Jim Davies
Computing Laboratory, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK

Abstract **Data sharing represents one of the key objectives and major challenges of today's cancer research. CancerGrid, a consortium of clinicians, cancer researchers, computational biologists and software engineers from leading UK institutions, is developing open-standards cancer informatics addressing this challenge. The CancerGrid solution involves the representation of a widely accepted clinical trials model in controlled vocabulary and common data elements (CDEs) as the enabling factor for cancer data sharing. This paper describes a cancer data query system that supports data sharing across CancerGrid-compliant clinical trial boundaries. The formal specification of the query system allows the model-driven development of a flexible, web-based interface that cancer researchers with limited IT experience can use to identify and query common data across multiple clinical trials.**

I. Introduction

CancerGrid [1] is a project that brings together clinicians, cancer researchers, bioinformaticians and software engineers from leading UK institutions. The project uses a generic clinical trials model [2] based on controlled vocabulary and common data elements (CDEs) [3] for the design, execution and analysis of cancer clinical trials. The advantage of this approach is twofold. Firstly, it makes possible the model-based development of open-standards IT systems for clinical trial execution [4]. Secondly, the CancerGrid approach enables data sharing across cancer clinical trial boundaries. The latter capability is demonstrated by the cross-trial query system presented in this paper.

The remainder of the paper is organised as follows. After a review of related work in the next section, Section III uses Z notation [5] to formally define common data elements. Section IV shows how CDE sets are associated to trial events to create a *trial design*, i.e., a full specification of a cancer clinical trial. Section V describes the cross-trial query model, and its use of the CDEs associated with the same execution stage (e.g., patient registration or follow-up) of multiple trials.

A proof-of-concept system that implements the cross-trial query model is presented in Section VI. The system allows cancer researchers with limited IT experience (e.g., clinicians and statisticians) to easily identify common data across multiple clinical trials, and to build queries targeted at these data using a familiar interface.

Section VII discusses two possible extensions of the query model. First, the grouping of CDEs associated with different trial execution stages is considered as a way of making queries less restrictive. Second, a solution to the generation of queries compliant with the security constraints specific to clinical trials is investigated. Section VIII concludes the paper with an overview of the query system, and an analysis of future work directions.

II. Related Work

The query of data from multiple sources has been an important research topic for the last two decades. Generic approaches for querying multiple information sources were proposed [6, 7, 8] that use a model of a problem domain to devise global query systems. The approach in [6] requires the user to build a semantic *domain model* as well as a model of each database and knowledge base used as an information source. Therefore, this solution is appropriate only for users with expertise in both data modelling and the target problem domain. Similar approaches are described in [7, 8], where sophisticated techniques are used to create a "metadatabase" [7] or a "reference data model" [8] that are then employed to generate the global query. Unlike these approaches that address the query of heterogeneous data sources, our query system takes advantage of the homogeneity of data across cancer clinical trials to hide most of the complexity of a cross-trial query. Implementations of this system can therefore be used directly by cancer researchers with limited data modelling expertise.

In the cancer research area, the US cancer Biomedical Informatics Grid (caBIG) project [9] models clinical trials [10] and has cancer data sharing as one of its primary objectives. Their caCORE software development kit [11] provides building blocks for many software components employed in cancer research. The inclusion of multiple data source querying in a proprietary language (i.e., the caBIG Query Language) is planned for the next release of the kit. While this will provide the functionality required to implement a system for querying multiple cancer data sources, the query system presented in this paper allows the automatic generation of a complete query form ready for immediate use by clinicians and statisticians.

Other medical projects such as VOTES [12] and PRATA [13] are concerned with the integration of data from multiple, distributed databases. The VOTES system [12] is concerned with the integration of distributed medical data pertaining to the same patient, so candidate patients for new clinical trials can be identified easily. The query forms used by the VOTES portal resemble those from the prototype implementation of the query system introduced in this paper, however they are encoded manually by software developers familiar with the internal structure of the data sources. The PRATA system [13] addresses the XML integration of data extracted from multiple, distributed databases. The integration and visualisation of the data is based on a user-specified XML schema that requires inside knowledge of the data sources. On the contrary, the CancerGrid query forms are model-based, and provide information to guide user querying rather than relying on the users for this knowledge.

III. Common Data Elements

The consistent use of a controlled vocabulary (i.e., a list of explicitly enumerated terms managed by a vocabulary registration authority) is key to sharing data between projects in any field of research. This is particularly relevant to cancer research, where tremendous human and financial resources are often employed for the generation of relatively small amounts of data [1]. The ability to analyse these data across multiple clinical trials is crucial to reaching statistically relevant conclusions.

The CancerGrid project is addressing this important requirement by basing its clinical trials model [2] on the use of thesauri—collections of controlled vocabulary terms and their relationships, and common data elements—controlled sets of cancer concepts and measurements. A common data element [3] is defined in terms of several basic types:

- *CdeID*, the set of all common data element identifiers. CDE identifiers are used to uniquely refer to specific CDEs.
- *CdeType*, the set of all types that CDE values may have. Typically, any XML schema simple type is allowed.
- *CdeInfo*, the actual details of the CDE, including a name and a description.

These basic types are summarised below using Z notation [5]:

$$[\mathit{CdeID}, \mathit{CdeType}, \mathit{CdeInfo}], \tag{1}$$

and the common data element type can be specified as:

$$
\begin{array}{|l}
\hline
\mathit{Cde} \\
\mathit{id} : \mathit{CdeID} \\
\mathit{valueDomain} : \mathit{CdeType} \\
\mathit{info} : \mathit{CdeInfo} \\
\hline
\end{array}
\tag{2}
$$

Common data elements used to model data in a specific research field are maintained in a CDE (or metadata) repository for that area of research:

$$
\begin{array}{|l}
\hline
\mathit{CdeRepository} \\
\mathit{cdeSet} : \mathbb{P}\, \mathit{Cde} \\
\hline
\forall\, x, y : \mathit{cdeSet} \bullet x.id = y.id \Rightarrow x = y \\
\hline
\end{array}
\tag{3}
$$

IV. Clinical Trial Events and Trial Designs

Clinical trial data are generated during the execution of a trial as a result of a number of trial events, each of which corresponds to a stage in the execution of the clinical trial. For instance, clinical and personal patient data are collected during the *registration* stage, treatments are allocated in the *randomisation* stage, and periodical *follow-up* data collection is performed to assess response to treatment. The complete set of trial events in the CancerGrid trial model is given below:

$$
\begin{aligned}
\mathit{TrialEvent} ::= {} & \mathit{registration} \mid \mathit{eligibility} \mid \mathit{randomisation} \mid \\
& \mathit{onStudy} \mid \mathit{treatment} \mid \mathit{offStudy} \mid \mathit{response} \mid \\
& \mathit{followUp} \mid \mathit{adverseEffect}
\end{aligned}
\tag{4}
$$

Clinicians gather the data corresponding to the trial events by filling in case report forms that comprise CDEs drawn from the cancer CDE repository [3],

$$\mid\; \mathit{cancerCdeRep} : \mathit{CdeRepository}. \tag{5}$$

For the purpose of data analysis, a clinical trial is composed of a set of trial events, each of which is associated with a set of common data elements [2]. This is defined by the *TrialDesign* specification below:

$$
\begin{array}{|l}
\hline
\mathit{TrialDesign} \\
\mathit{events} : \mathbb{P}\, \mathit{TrialEvent} \\
\mathit{eventCdeSet} : \mathit{TrialEvent} \mathrel{+\!\!\!\!\!\rightarrow} \mathbb{P}\, \mathit{cancerCdeRep.cdeSet} \\
\hline
\operatorname{dom} \mathit{eventCdeSet} = \mathit{events} \\
\hline
\end{array}
\tag{6}
$$

To give an example of a clinical trial design, consider the following common data elements:

$$
\begin{array}{|l}
\mathit{NodalStatus}, \mathit{AdjuvantRadiotherapy}, \mathit{Her2Level}, \\
\mathit{ECOGStatus}, \mathit{InvasiveCarcinoma}, \mathit{TumorResectionStatus}, \\
\mathit{DiseaseStage}, \mathit{AdjuvantChemotherapyIndication}, \\
\mathit{PatientFitness}, \mathit{BoneMarrowHepaticRenalFunction}, \\
\mathit{InformedConsent}, \mathit{NoPreviousTherapy}, \mathit{KnownRadiotherapy}, \\
\mathit{LastSurgeryDate}, \mathit{NoPreviousMalignancy}, \\
\mathit{NotPregnantLactating}, \mathit{PatientNameInitials}, \\
\mathit{PatientBirthDate}, \mathit{TissueSubstudyConsent}, \\
\mathit{QualityOfLifeSubstudyConsent}, \mathit{ProgesteroneReceptorStatus}, \\
\mathit{OestrogenReceptorStatus} : \mathit{cancerCdeRep.cdeSet}
\end{array}
\tag{7}
$$

that are associated with three of the trial events for the tAnGo clinical trial [14]:

$$
\begin{array}{|l}
\mathit{tAnGo} : \mathit{TrialDesign} \\
\hline
\{\mathit{registration}, \mathit{eligibility}, \mathit{randomisation}\} \subset \mathit{tAnGo.events} \\
\mathit{tAnGo.eventCdeSet}\ \mathit{registration} = \\
\quad \{\mathit{TissueSubstudyConsent}, \mathit{QualityOfLifeSubstudyConsent}, \\
\quad \mathit{ProgesteroneReceptorStatus}, \mathit{OestrogenReceptorStatus}\}. \\
\mathit{tAnGo.eventCdeSet}\ \mathit{eligibility} = \\
\quad \{\mathit{InvasiveCarcinoma}, \mathit{TumorResectionStatus}, \\
\quad \mathit{DiseaseStage}, \mathit{AdjuvantChemotherapyIndication}, \\
\quad \mathit{PatientFitness}, \mathit{BoneMarrowHepaticRenalFunction}, \\
\quad \mathit{InformedConsent}, \mathit{NoPreviousTherapy}, \\
\quad \mathit{KnownRadiotherapy}, \mathit{LastSurgeryDate}, \\
\quad \mathit{NoPreviousMalignancy}, \mathit{NotPregnantLactating}\} \\
\mathit{tAnGo.eventCdeSet}\ \mathit{randomisation} = \\
\quad \{\mathit{NodalStatus}, \mathit{AdjuvantRadiotherapy}, \mathit{Her2Level}, \\
\quad \mathit{ECOGStatus}\}
\end{array}
\tag{8}
$$

The common data elements used to register *tAnGo* participants, to establish their eligibility, and to stratify the allocation of treatments for the eligible participants (i.e., the trial *randomisation* [15]) are explicitly specified in the *tAnGo* trial design. Note that the complete trial design for tAnGo comprises all of the trial events defined in (4), however for the sake of brevity only three of these are presented above.

V. Clinical Trial Queries

Having introduced the data components of a CancerGrid clinical trial in the previous section, we will now define the cross-trial queries for sharing data among clinical trials using the same CDE repository. A number of comparison operators are used to build the query:

$$\begin{aligned}&\mathit{ComparisonOp} ::= \mathit{hasAnyValue} \mid \mathit{isEqualTo} \mid \\ &\quad \mathit{isNotEqualTo} \mid \mathit{isLessThan} \mid \mathit{isGreaterThan} \mid \\ &\quad \mathit{isLessThanOrEqualTo} \mid \mathit{isGreaterThanOrEqualTo}.\end{aligned} \tag{9}$$

Each *CdeType* type that CDEs can draw their values from is associated with a well-defined set of these operators:

$$\mathit{cdeComparisonOp} : \mathit{CdeType} \rightarrow \mathbb{P}\ \mathit{ComparisonOp}. \tag{10}$$

For our purpose, a query element comprises a CDE and a comparison operator that is relevant for its value domain:

$$\begin{array}{|l}\hline \mathit{QueryElement} \\ \mathit{cde} : \mathit{Cde} \\ \mathit{op} : \mathit{ComparisonOp} \\ \hline \mathit{op} \in \mathit{cdeComparisonOp}\ \mathit{cde}.\mathit{valueDomain} \\ \hline \end{array} \tag{11}$$

The query system described in this section is event based, namely we are interested in identifying CDEs that are associated with the same trial event in all clinical trials involved in the query. This approach is consistent with the cancer researchers' need to analyse data from patients with similar characteristics at the same stage of their treatment. For instance, it makes sense to group patient data collected prior to the commencement of treatment for trials where the treatment varies, because this will avoid the confounding effects of the different treatments under study. However, comparisons at later time points may prove less useful.

Given a set of clinical trials *trialSet*, the query system specifies its query terms as a mapping from trial events to sets of query elements:

$$\begin{array}{|l}\hline \mathit{TrialQuery} \\ \mathit{trialSet} : \mathbb{P}\ \mathit{TrialDesign} \\ \mathit{queryTerms} : \mathit{TrialEvent} \rightarrow\!\!\!\!\!\!+\ \mathbb{P}\ \mathit{QueryElement} \\ \hline \mathrm{dom}\ \mathit{queryTerms} = \bigcap\{t : \mathit{trialSet} \bullet t.\mathit{events}\} \\ \forall\, e : \mathrm{dom}\ \mathit{queryTerms} \bullet \{\mathit{qt} : \mathit{queryTerms}\ e \bullet \mathit{qt}.\mathit{cde}\} = \\ \quad \bigcap\{t : \mathit{trialSet} \bullet t.\mathit{eventCdeSet}\ e\} \\ \forall\, e : \mathrm{dom}\ \mathit{queryTerms} \bullet \forall\, \mathit{qt}1, \mathit{qt}2 : \mathit{queryTerms}\ e \bullet \\ \quad \mathit{qt}1 \neq \mathit{qt}2 \Rightarrow \mathit{qt}1.\mathit{cde} \neq \mathit{qt}2.\mathit{cde} \\ \hline \end{array} \tag{12}$$

The first constraint in the query definition requires that the trial events involved in the query must be part of all considered trials.[1] The last two constraints state that the query terms for these events comprise precisely one *QueryElement*[2] for every CDE that the event is associated with in each of the queried trials. Notice that the *TrialQuery* specification in (12) does not place any restriction on the comparison operators that are part of the *QueryElement* items associated with trial events. The only such constraint is specified by the definition of a *QueryElement* (11), i.e., these operators must be appropriate for the CDEs they relate to.

To illustrate the application of the trial query, consider the NEAT clinical trial [16], which uses several additional CDEs alongside those introduced in the previous section:

$$\begin{array}{|l} \mathit{RadiotheraphyTiming}, \mathit{TumorSize}, \mathit{TumorGrade}, \\ \mathit{CyclophospamidePlan}, \mathit{MenopausalStatus}, \\ \mathit{TamoxifenPlan} : \mathit{cancerCdeRep}.\mathit{cdeSet} \\ \mathit{Neat} : \mathit{TrialDesign} \\ \hline \{\mathit{registration}, \mathit{eligibility}, \mathit{randomisation}\} \subset \mathit{Neat}.\mathit{events} \\ \mathit{Neat}.\mathit{eventCdeSet}\ \mathit{registration} = \\ \quad \{\mathit{QualityOfLifeSubstudyConsent}, \\ \quad \mathit{OestrogenReceptorStatus}, \mathit{TumorSize}, \\ \quad \mathit{TumorGrade}, \mathit{ECOGStatus}, \mathit{CyclophospamidePlan}, \\ \quad \mathit{MenopausalStatus}, \mathit{TamoxifenPlan}\} \\ \mathit{Neat}.\mathit{eventCdeSet}\ \mathit{eligibility} = \\ \quad \{\mathit{InvasiveCarcinoma}, \mathit{DiseaseStage}, \\ \quad \mathit{TumorResectionStatus}, \mathit{AdjuvantChemotherapyIndication}, \\ \quad \mathit{PatientFitness}, \mathit{InformedConsent}, \\ \quad \mathit{BoneMarrowHepaticRenalFunction}, \\ \quad \mathit{NoPreviousMalignancy}, \mathit{NotPregnantLactating}\} \\ \mathit{Neat}.\mathit{eventCdeSet}\ \mathit{randomisation} = \\ \quad \{\mathit{NodalStatus}, \mathit{RadiotheraphyTiming}\} \end{array} \tag{13}$$

An event-based query across the tAnGo trial in (8) and the NEAT trial above is then given by:

$$\begin{array}{|l} \mathit{tAnGoNeatQuery} : \mathit{TrialQuery} \\ \hline \mathit{tAnGoNeatQuery}.\mathit{trialSet} = \{\mathit{tAnGo}, \mathit{Neat}\}. \end{array} \tag{14}$$

According to the definition of a *TrialQuery* in (12),

$$\begin{gathered}\{\mathit{registration}, \mathit{eligibility}, \mathit{randomisation}\} \subseteq \\ \mathrm{dom}\ \mathit{tAnGoNeatQuery}.\mathit{queryTerms}\end{gathered} \tag{15}$$

since all these trial events appear in both *tAnGo* and *Neat*. The CDE sets that are part of the *queryTerms* (12) for the three trial events are:

$$\begin{aligned}&\{\mathit{qt} : \mathit{tAnGoNeatQuery}.\mathit{queryTerms}\ \mathit{randomisation} \bullet \mathit{qt}.\mathit{cde}\} = \\ &\quad \{\mathit{NodalStatus}\} \\ &\{\mathit{qt} : \mathit{tAnGoNeatQuery}.\mathit{queryTerms}\ \mathit{eligibility} \bullet \mathit{qt}.\mathit{cde}\} = \\ &\quad \{\mathit{InvasiveCarcinoma}, \mathit{TumorResectionStatus}, \\ &\quad \mathit{DiseaseStage}, \mathit{AdjuvantChemotherapyIndication}, \\ &\quad \mathit{PatientFitness}, \mathit{BoneMarrowHepaticRenalFunction}, \\ &\quad \mathit{InformedConsent}, \mathit{NoPreviousMalignancy}, \\ &\quad \mathit{NotPregnantLactating}\} \\ &\{\mathit{qt} : \mathit{tAnGoNeatQuery}.\mathit{queryTerms}\ \mathit{registration} \bullet \mathit{qt}.\mathit{cde}\} = \\ &\quad \{\mathit{QualityOfLifeSubstudyConsent}, \\ &\quad \mathit{OestrogenReceptorStatus}\}.\end{aligned} \tag{16}$$

Appropriate comparison operators are associated with each of these CDEs in the above *TrialQuery* instance, e.g.,

$$\begin{aligned}&\mathit{tAnGoNeatQuery}.\mathit{queryTerms}\ \mathit{randomisation} = \\ &\qquad \{\langle\!|\, \mathit{cde} \leadsto \mathit{NodalStatus}, \mathit{op} \leadsto \mathit{isEqualTo} \,|\!\rangle\}\end{aligned} \tag{17}$$

Prior to being executed, a *TrialQuery* instance such as *tAnGoNeatQuery* needs to be parameterised by a set of values from the value domains of all CDEs in the query terms. For the *tAnGoNeatQuery* query term in (17) for instance, *op* was chosen to be *isEqualTo*, so the *NodalStatus* value of interest will need to be specified in an implementation of the query framework. This part of the query is not modelled here, however details are provided in the next section that presents a proof-of-concept realisation of the system.

[1] This simplifying assumption is relaxed in Section VII, which proposes a generalisation of the basic cross-trial query in (12).

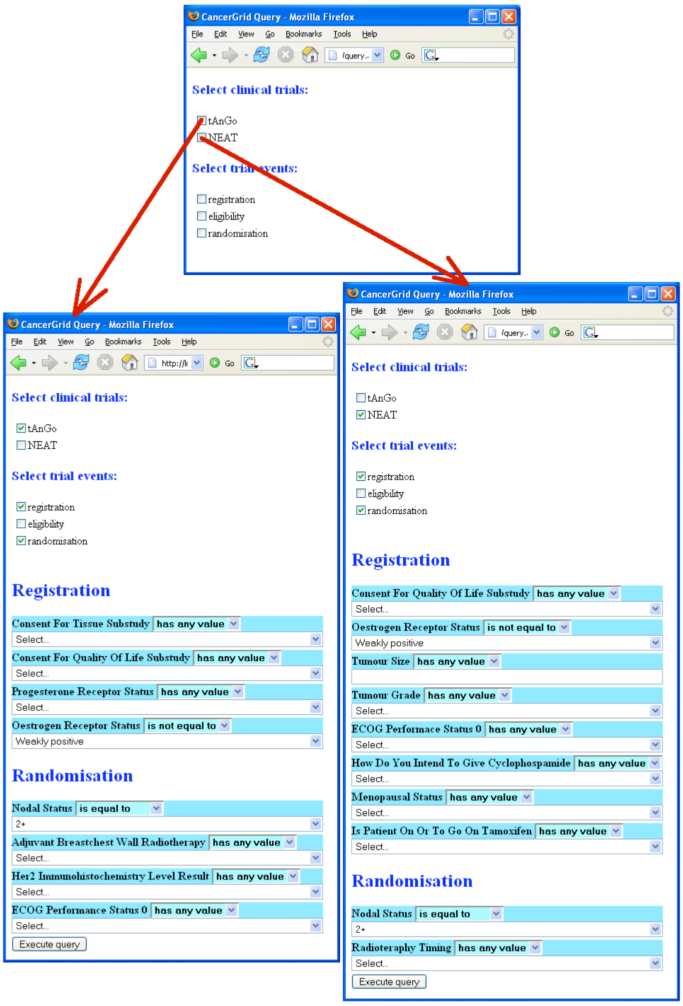
[2] Although each of the *queryTerms* consists of a single *QueryElement*, the query model can be extended easily to handle terms defined as logical expressions of multiple *QueryElements* that refer to the same CDE.

VI. Case Study

The cross-trial query system introduced in the previous section was applied to two of the primary CancerGrid clinical trials, tAnGo [14] and NEAT [16]. The Microsoft ASP.NET platform [17] was chosen for the implementation of the query system prototype. This choice was motivated by a range of factors including customer acceptance, simplicity, and the ability to reuse existing model-based web form generators that CancerGrid developed for the automatic generation of service-oriented architectures (SOAs) for clinical trial execution [4].

The reuse of the CancerGrid SOA generation components ensures that changes and enhancements to the query system can be easily converted into query web forms. Additionally, this approach provides a query user interface that clinical trial personnel will already be familiar with after having filled in similar forms during the actual execution of the trial.

Fig.1 shows the query web form generated for the two clinical trials. As shown at the top of Fig. 1, in the degenerate case when an empty *trialSet* is used, the *queryTerms* set is itself empty:



CancerGrid Query - Mozilla Firefox

Select clinical trials:
□ tAnGo
□ NEAT

Select trial events:
□ registration
□ eligibility
□ randomisation

CancerGrid Query - Mozilla Firefox

Select clinical trials:
☑ tAnGo
□ NEAT

Select trial events:
☑ registration
□ eligibility
☑ randomisation

Registration

Consent For Tissue Substudy: has any value; Select...
Consent For Quality Of Life Substudy: has any value; Select...
Progesterone Receptor Status: has any value; Select...
Oestrogen Receptor Status: is not equal to; Weakly positive

Randomisation

Nodal Status: is equal to; 2+
Adjuvant Breastchest Wall Radiotherapy: has any value; Select...
Her2 Immunohistochemistry Level Result: has any value; Select...
ECOG Performance Status 0: has any value; Select...

Execute query

CancerGrid Query - Mozilla Firefox

Select clinical trials:
□ tAnGo
☑ NEAT

Select trial events:
☑ registration
□ eligibility
☑ randomisation

Registration

Consent For Quality Of Life Substudy: has any value; Select...
Oestrogen Receptor Status: is not equal to; Weakly positive
Tumour Size: has any value
Tumour Grade: has any value; Select...
ECOG Performace Status 0: has any value; Select...
How Do You Intend To Give Cyclophospamide: has any value; Select...
Menopausal Status: has any value; Select...
Is Patient On Or To Go On Tamoxifen: has any value; Select...

Randomisation

Nodal Status: is equal to; 2+
Radioteraphy Timing: has any value; Select...

Execute query

Fig. 1. Single-trial queries as a special case of the cross-trial query system. The familiar, easy-to-use web form allows cancer researchers with limited IT experience to identify and query common data across multiple clinical trials, and have their query automatically encoded into a set of dedicated SQL statements, one for each of the trials.

$$\begin{array}{|l} emptyQuery : TrialQuery \\ \hline emptyQuery.trialSet = \emptyset \end{array} \qquad (18)$$

By adding either trial to the *trialSet*, an ordinary, single-trial query system is obtained (Fig. 1), e.g.,

$$\begin{array}{|l} tAnGoQuery : TrialQuery \\ \hline tAnGoQuery.trialSet = \{tAnGo\}. \end{array} \qquad (19)$$

Although this use case does not involve any data sharing, the ability to perform ordinary trial queries using the same system is useful. The power of the cross-trial query technique is put to use by including both the *tAnGo* and the *Neat* clinical trials into the *trialSet*, as defined by *tAnGoNeatQuery* in (14) and illustrated in Fig. 2.

VII. Query System Extensions

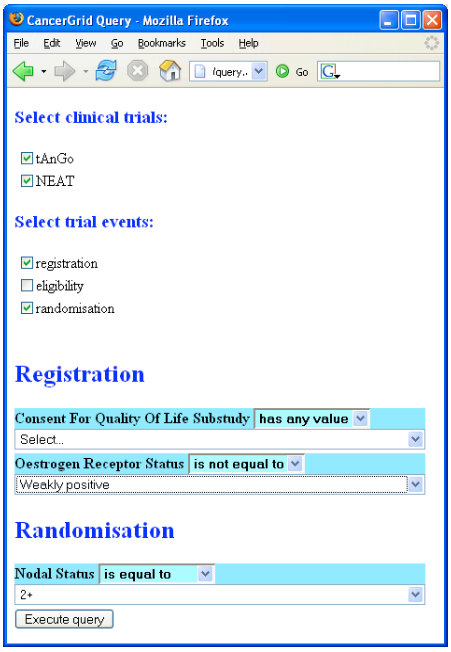
A. Trial event combining

Common data elements used in different clinical trials are not necessarily associated with the same trial event in each of these trials. An example is the *ECOGStatus* CDE defined in (7), which is used by both trials considered in the previous sections:

$$ECOGStatus \in tAnGo.eventCdeSet\ randomisation \qquad (20)$$

but

$$ECOGStatus \in Neat.eventCdeSet\ registration, \qquad (21)$$



CancerGrid Query - Mozilla Firefox

Select clinical trials:
☑ tAnGo
☑ NEAT

Select trial events:
☑ registration
□ eligibility
☑ randomisation

Registration

Consent For Quality Of Life Substudy: has any value; Select...
Oestrogen Receptor Status: is not equal to; Weakly positive

Randomisation

Nodal Status: is equal to; 2+

Execute query

Fig. 2. The *tAnGoNeatQuery* cross-trial query, with two trial events selected. The sets of query terms for the randomisation and registration trial events are $tAnGoNeatQuery.queryTerms\ randomisation = \{\langle\!| cde \leadsto NodalStatus, op \leadsto isEqualTo |\!\rangle\}$ and $tAnGoNeatQuery.queryTerms\ registration = \{\langle\!| cde \leadsto QualityOfLifeSubstudyConsent, op \leadsto hasAnyValue |\!\rangle, \langle\!| cde \leadsto OestrogenReceptorStatus, op \leadsto isNotEqualTo |\!\rangle\}$, respectively.

In both cases the measurement is taken prior to patients joining the clinical trial, so the ability to perform queries including query terms for such CDEs is very important.

Relaxing the requirement that cross-trial queries are event-based (in the sense described in the previous section) opens up a whole spectrum of possibilities. At one end of this spectrum, trial events can be completely disregarded and queries can be performed on the intersection of the entire CDE sets across the considered trials. At the other end of the spectrum, a user-specified CDE grouping across trials can be used.

The usefulness of either approach is limited by their complexity, so our query generalisation will focus on combining trial events into subsets and using these subsets as the basis for the query generation. This approach addresses the use case mentioned at the beginning of this section, and is both simple and effective in many real-world scenarios.

The generalised cross-trial query is very similar to the basic trial query, except that event sets rather than individual events are mapped on to query elements:

$$\begin{array}{|l}
\textit{GeneralisedTrialQuery} \\
\hline
\textit{trialSet} : \mathbb{P}\ \textit{TrialDesign} \\
\textit{queryTerms} : \mathbb{P}\ \textit{TrialEvent} \nrightarrow \mathbb{P}\ \textit{QueryElement} \\
\hline
\bigcup(\mathrm{dom}\ \textit{queryTerms}) = \bigcap\{t : \textit{trialSet} \bullet t.\textit{events}\} \\
\forall\ \textit{events} : \mathrm{dom}\ \textit{queryTerms} \bullet \\
\quad\quad \{\textit{qt} : \textit{queryTerms events} \bullet \textit{qt.cde}\} = \\
\quad\quad\quad \bigcap\{t : \textit{trialSet} \bullet \bigcup\{e : \textit{events} \bullet t.\textit{eventCdeSet}\ e\}\} \\
\forall\ \textit{events} : \mathrm{dom}\ \textit{queryTerms} \bullet \forall\ \textit{qt}1, \textit{qt}2 : \textit{queryTerms events} \bullet \\
\quad\quad \textit{qt}1 \neq \textit{qt}2 \Rightarrow \textit{qt}1.\textit{cde} \neq \textit{qt}2.\textit{cde} \\
\hline
\end{array} \quad (22)$$

The change from individual events to event sets in the definition of the *queryTerms* component of the query led to a couple of changes to the constraints. The first one relaxes the constraint on the domain of *queryTerms*, specifying it as any set of event sets whose union gives the whole set of overlapping events for the considered trials. The change to the second constraint specifies that the CDE components of the query elements associated with a trial event set are obtained by considering the CDEs associated with all the events in the set.

Notice that the above definition does not require that different event sets involved in the query are disjoint. If this is required, then the domain of the *queryTerms* mapping becomes a partition of the *queryTerms* domain, and the query is specialised to:

$$\begin{array}{l}
\textit{PartitionedGeneralisedTrialQuery} \mathrel{\widehat=} [\textit{GeneralisedTrialQuery} \mid \\
\quad\quad \forall\ x, y : \mathrm{dom}\ \textit{queryTerms} \bullet x \cap y \neq \emptyset \Rightarrow x = y]
\end{array} \quad (23)$$

If we further constrain the query in (23) so that each element of the partition contains precisely one event, we obtain the *singleton* query below:

$$\begin{array}{l}
\textit{SingletonGeneralisedTrialQuery} \mathrel{\widehat=} [\textit{PartitionedTrialQuery} \mid \\
\quad\quad \forall\ x : \mathrm{dom}\ \textit{queryTerms} \bullet \#x = 1].
\end{array} \quad (24)$$

This is equivalent to the original trial query in (12).[3]

B. Access Control

The query system needs to be security aware, so that access to confidential information is limited to the rightful users. Clinical trials use a role-based access control (RBAC) approach [18, 19] to constrain data access to users that have certain roles in the trial, and our query system needs to reflect this. Given the set of possible user roles

$$[\textit{Role}], \quad (25)$$

a RBAC-aware trial design defines the rules that specify the common data elements that each role can query :

$$\begin{array}{|l}
\textit{RbacTrialDesign} \\
\hline
\textit{trial} : \textit{TrialDesign} \\
\textit{accessRules} : \textit{Role} \rightarrow \mathbb{P}\ \textit{Cde} \\
\hline
\forall\ \textit{roleCdeSet} : \mathrm{ran}\ \textit{accessRules} \bullet \\
\quad\quad \textit{roleCdeSet} \subseteq \bigcup\{e : \textit{trial.events} \bullet \textit{trial.eventCdeSet}\ e\} \\
\hline
\end{array} \quad (26)$$

Users that access data corresponding to an RBAC-enabled clinical trial have a view of the clinical trial that is specific to their roles. For the purpose of querying trial data, each of these customised trial views is identical to the view provided by a basic *TrialDesign* instance obtained by filtering out the inaccessible CDEs from the original trial. The following filter maps a (*Role*, *RbacTrialDesign*) pair to its query-equivalent basic trial:

$$\begin{array}{|l}
\textit{filter} : (\textit{Role} \times \textit{RbacTrialDesign}) \rightarrow \textit{TrialDesign} \\
\hline
\forall\ r : \textit{Role} \bullet \forall\ t : \textit{RbacTrialDesign} \bullet \\
\quad\quad (\textit{filter}\ (r, t)).\textit{events} = t.\textit{trial.events} \\
\forall\ r : \textit{Role} \bullet \forall\ t : \textit{RbacTrialDesign} \bullet \forall\ e : (\textit{filter}\ (r, t)).\textit{events} \bullet \\
\quad\quad (\textit{filter}\ (r, t)).\textit{eventCdeSet}\ e = \\
\quad\quad\quad\quad t.\textit{trial.eventCdeSet}\ e \cap t.\textit{accessRules}\ r
\end{array} \quad (27)$$

The constraint part of the filter definition describes how the two components of a basic *TrialDesign* (i.e., its event set and its event-CDEs associations) are built from the components of the RBAC-enabled trial. Notice that while the basic trial has the same event set as the RBAC-enabled trial, the filtered event-to-CDEs mappings are obtained by eliminating all CDEs inaccessible to the considered role from the original trial.

Given a set of RBAC-enabled trials and a set of associated user roles[4], a cross-trial query can be obtained by first building the set of query-equivalent basic trials using the *filter* in (27), and then employing a *TrialQuery* for the "filtered" trials. This process is described below:

$$\begin{array}{|l}
\textit{RbacTrialQuery} \\
\hline
\textit{trialSet} : \mathbb{P}\ \textit{RbacTrialDesign} \\
\textit{roles} : \textit{RbacTrialDesign} \nrightarrow \textit{Role} \\
\textit{queryTerms} : \textit{TrialEvent} \nrightarrow \mathbb{P}\ \textit{QueryElement} \\
\hline
\mathrm{dom}\ \textit{roles} = \textit{trialSet} \\
\exists\ q : \textit{TrialQuery} \bullet q.\textit{trialSet} = \{t : \textit{trialSet} \bullet \textit{filter}(\textit{roles}\ t, t)\} \wedge \\
\quad\quad\quad q.\textit{queryTerms} = \textit{queryTerms} \\
\hline
\end{array} \quad (28)$$

[3] A bijection can be defined that maps each *SingletonGeneralisedTrialQuery* to a *TrialQuery*.

[4] Note that the same user can have different roles for each of the trials, hence the use of a set of roles rather than a single role across all clinical trials.

The definition of the *filter* operator in (27) ensures that an *RbacTrialQuery* will include all CDEs accessible to the user with the specified *Roles* and no other CDE.

VIII. Conclusions and Further Work

The consistent use of controlled vocabulary and common data elements [3] creates the potential for data sharing in cancer clinical trials [1]. The query system described in this paper realizes this potential by proving cancer researchers and statisticians with a straightforward means for accessing data across multiple clinical trials.

The basic CancerGrid query system offers complex queries targeted at the CDEs associated with the same trial event for each clinical trial of interest. The extensions presented in the previous section allow users to customize the system so that flexible queries extending across trial event boundaries are possible with minimal configuration effort, and the strict security requirements of clinical trials are thoroughly addressed.

Further work is required to validate the proposed extensions with their user community, and to complete an implementation of the system that provides these extensions. An important characteristic of this implementation will be the ability to dynamically include into the query trial set all trials that the user has the authority to access, with the appropriate queries generated at runtime. A further direction to be investigated is the modelling of queries including elements that refer to CDEs that do not appear in all of the queried trials.

Another cross-trial query approach that CancerGrid is considering involves the use of semantic reasoners [20], and additional work is required to combine this approach with the one described in the current paper. Finally, it is our plan to extend and generalize the cross-project query system to other domains that would benefit from CDE-based data sharing within the respective research fields.

Acknowledgements

This work was supported by the UK Medical Research Council grant G0300648. The authors are grateful to Peter Maccallum, Charlie Crichton, Andrew Tsui and the other members of the CancerGrid team for many insightful discussions during the development of the cross-trial query system.

References

[1] James Brenton, Carlos Caldas, Jim Davies, Steve Harris and Peter Maccallum, "CancerGrid: developing open standards for clinical cancer informatics", Proceedings of the UK e-Science All Hands Meeting 2005, pp. 678-681, http://www.allhands.org.uk/2005/proceedings/.

[2] Steve Harris and Radu Calinescu, "Clinical trials model 1.0", CancerGrid Technical Report MRC/1.4.1.1, June 2006, http://www.cancergrid.org/public/documents/2006/mrc/Report%20MRC-1.4.1.1%20Clinical%20trials%20model%201.0.pdf.

[3] Igor Toujilov and Peter Maccallum, "Common data element management architecture", CancerGrid Technical Report MRC-1.1.2, May 2006, http://www.cancergrid.org/public/documents/2006/mrc/Report%20MRC-1.1.2%20CDE%20management%20architecture.pdf**.**

[4] Radu Calinescu, "Model-based SOA generation for cancer clinical trials", Best Practices and Methodologies in Service-Oriented Architectures – Proceedings of the 4th OOPSLA International Workshop on SOA and Web Services, Portland, October 2006, pp. 57-71.

[5] Jim Woodcock and Jim Davies, "Using Z. Specification, Refinement and Proof", Prentice Hall, 1996.

[6] Yigal Arens, Chin Y. Chee, Chun-Nan Hsu and Craig A. KnoBlock, "Retrieving and Integrating Data from Multiple Information Sources", Journal of Intelligent and Cooperative Information Systems, vol. 2, no. 2, June 1993, pp. 127-158.

[7] Waiman Cheung and Cheng Hsu, "The Model-Assisted Global Query System for Multiple Databases in Distributed Enterprises", ACM Transactions on Information Systems, vol. 14, no. 4, October 1996, pp. 421–470.

[8] Silvana Castano, Valeria De Antonellis and Sabrina De Capitani di Vimercati, "Global Viewing of Heterogeneous Data Sources", IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 2, March/April 2001, pp. 277-297.

[9] US National Cancer Institute, The cancer Biomedical Informatics Grid, 2006, https://cabig.nci.nih.gov/.

[10] Rebecca Kush, "Can the Protocol be Standardised?", Clinical Data Interchange Standards Consortium, 2006, http://www.cdisc.org/publications/CDISK_ed.pdf.

[11] Peter A. Covitz, Frank Hartel, Carl Schaefer, Sherri De Corondao, Gilberto Fragoso, Himanso Sahni et al., "caCORE: A common infrastructure for cancer informatics", Bioinformatics, vol. 19, no. 18, pp. 2404-2412, 2003, http://bioinformatics.oxfordjournals.org/cgi/content/abstract/19/18/2404.

[12] Anthony Stell, Richard Sinnott and Oluwafemi Ajayi, "Supporting the Clinical Trial Recruitment Process through the Grid", Proceedings of the UK e-Science All Hands Meeting 2006, pp. 61-68.

[13] Gao Cong, Wenfei Fan, Xibei Jia and Shuai Ma, "PRATA: A System for XML Publishing, Integration and View Maintenance", Proceedings of the UK e-Science All Hands Meeting 2006, pp. 432-435.

[14] Chris Poole, Helen Howard and Janet Dunn, "tAnGo — a randomised phase III trial of gemcitabine in paclitaxel-containing, epirubicin-based, adjuvant chemotherapy for women with early stage breast cancer", 2003, http://www.isdscotland.org/isd/files/tAnGo_protocol_version_2.0_July_2003.pdf.

[15] Elaine Beller, Val Gebski and Anthony Keech, "Randomisation in clinical trials", Medical Journal of Australia, vol. 177, Nov. 2002, pp. 565-567, http://www.mja.com.au/public/issues/177_10_181102/bel10697_fm.pdf.

[16] Earl H M, Poole C J, Dunn J A, Hiller L, Bathers S, Spooner D et al, "NEAT - National Epirubicin Adjuvant Trial, a multi-centre phase III randomised trial of Epirubicin x 4 and classical CMFx4 [ECMF] versus CMFx6", Proceedings of the American Society of Clinical Oncology **21** (2), 2002, pp. 1081-0641.

[17] Bill Evjen, Scott Hanselman, Farhan Muhammad, Srinivasa Sivakumar and Devin Rader, Professional ASP.NET 2.0, Wiley Publishing, 2006.

[18] D.F.Ferraiolo, D.R.Kuhn and R.Chandramouli, Role-Based Access Control, Computer Security Series, Artech House, 2003.

[19] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn and Ramaswamy Chandramouli, "Proposed NIST standard for role-based access control", ACM Transactions on Information and System Security, 4(3), pp. 224–274, August 2001, http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf.

[20] V. Haarslev and R. Moller, "Racer system description", Proceedings of the First International Joint Conference on Automated Reasoning, Lecture Notes in Computer Science, vol. 2083, Springer-Verlag, 2001, pp. 701-706.