The CancerGrid Experience: Metadata-Based Model-Driven Engineering for Clinical Trials

Jim Davies, Jeremy Gibbons, Steve Harris, Charles Crichton

Department of Computer Science, University of Oxford

Abstract

The CancerGrid approach to software support for clinical trials is based on two principles: careful curation of semantic metadata about clinical observations, to enable subsequent data integration, and model-driven generation of trial-specific software artifacts from a trial protocol, to streamline the software development process. This paper explains the approach, presents four varied case studies, and discusses the lessons learned.

Keywords: Metadata, ISO/IEC 11179, semantic frameworks, model-driven engineering, clinical informatics, electronic government

1. Introduction

1.1. Background

Randomized controlled trials are considered to be the 'gold standard' for experiments in medicine. They provide the most reliable evidence supporting or refuting a scientific hypothesis, such as that 'treatment *X* cures more patients suffering from disease *D* than does treatment *Y*'. An experiment is designed: treatment regimes *X* and *Y* will be specified; patients suffering from disease *D* will be recruited; recruits will be stratified into groups with similar relevant characteristics, based on factors such as age, gender and lifestyle; patients within each group will be allocated at random to treatment *X* or treatment *Y*; the results will be analyzed to determine whether or not the difference in effect of the treatments is statistically significant.

From a software point of view, a clinical trial is largely an exercise in data management: observations have to be specified, collected, recorded, integrated, and analyzed. But the software engineering aspects of setting up and running a clinical trial are not trivial. Two particular problems that we will address in this paper involve *data integration* and *tool generation*.

The data integration problem occurs because medical researchers want to be able to combine the results of multiple trials, a process known as *meta-analysis*. It is often the case that a single trial in isolation does not have adequate statistical power to yield a robustly significant conclusion. Nevertheless, if sufficiently many trials have been conducted, investigating sufficiently similar hypotheses and collecting sufficiently similar data, it may be possible to pool the results to greater effect. In other situations, the meta-analysis aims at evaluating new hypotheses that are formulated long after the completion of the trials that originally collected the data involved—in this case, data from trials investigating quite different hypotheses may be integrated.

Either way, for meta-analysis to be possible, it is necessary to capture and curate metadata expressing the 'semantics' of the data—only then is it possible to determine whether data collected in different trials are commensurate, and if so, how to relate them. For example, when measuring blood pressure, it is not enough to record a pair of numbers, or a pair of pressures, or a pair of measurements in mmHg, or even to indicate that these represent systolic and diastolic pressure. It is also necessary to know how that data was collected (at rest, or after five minutes on a treadmill?), and maybe factors such as who collected it (in the clinic by a professional, or at home by the patient?) and how recent

Email address: `firstname.lastname@cs.ox.ac.uk` (Jim Davies, Jeremy Gibbons, Steve Harris, Charles Crichton)

and reliable it is. This semantic metadata is an essential part of the context of the data, and logically forms part of the model of the trial, alongside more syntactic metadata such as the name of the trial and the types of data items.

As for tool development, current standard practice in clinical trials management is to pass a textual document containing the trial protocol over to database programmers in the clinical trials unit, or to consultants from a trials management software provider, who will use it as guidance in manually configuring an information management system for this particular trial. This practice causes numerous problems. Firstly, it induces delays: it is usually the case that some to-ing and fro-ing is needed between the database programmers and the medical researchers to get the details right; but the medics are often too busy to respond immediately, and it is not uncommon for the trial to have to start on paper because the software is not ready. Secondly, it is costly. This is not such a problem for a big 'phase III' trials operated on behalf of pharmaceutical companies pursuing regulatory approval: the study will have thousands of participants and a stable design, so the software development will form only a small proportion of the overall cost, and is likely to be recouped in sales over the lifetime of the drug. However, it is a problem for early-phase exploratory studies and late-phase post-approval studies: the former are much smaller, more dynamic and inherently risky, as animal models are an unreliable predictor of efficacy in humans; the latter are typically funded by charities, governments and NGOs in academic settings on a tight budget. Even then, many promising drugs are not brought to market because the return on the drug outweighs the cost of approval. Thirdly, it is not uncommon for an early-phase trial protocol to undergo changes during the execution of the trial, requiring adjustments to software components of the associated trial management system. Current practice is to implement these changes through manually modifying the underlying code, running the risk of introducing software bugs when the system is in production use. And finally, bespoke database design on a per-trial basis is unlikely to promote the consistency and interoperability needed for meta-analysis.

All four of these generation issues could be addressed if the development of the software tools needed to support trial execution could be automated. Fortunately, there is essentially enough information in the trial protocol—which needs to be written anyway, not least for the purposes of regulatory approval—to completely determine the relevant software artifacts, either from scratch or by configuring more generic components. If the protocol were written in a more structured format—that is, as a formal model, rather than merely a textual description, of the trial—then both the prose and the code could be generated from it by suitable processing, and any adjustments required because of changes to the trial protocol can be made without risky manual intervention at the level of code. Moreover, as we have seen, the annotation of the data descriptions in the trial model with semantic metadata will make that model doubly useful, as a basis for supporting meta-analysis in addition to being a specification for a software system.

In other words, clinical trials management is crying out for a model-driven approach.

1.2. The CancerGrid approach, in a nutshell

The CancerGrid project [1] was initiated in order to address the twin problems of interoperability and generativity in clinical trials, taking a model-driven approach to the development of trials management tools. It was funded in the first instance for three years from 2005 by the UK Medical Research Council, with the involvement of five UK universities: Cambridge (specializing in oncology), Oxford (software engineering), University College London (semantic modelling), Birmingham (clinical trials management), and Belfast (telemedicine). Oxford University and the Cancer Research UK Cambridge Research Institute have been continuing the work since the original project ended in 2008.

The CancerGrid approach addresses the two problems of data integration and tool generation, via the collection and management of metadata in the first case, and model-driven engineering in the second—improving the science through greater effectiveness, and reducing drudgery through greater efficiency.

Regarding the metadata, much of the interoperability requirement pivots on some kind of consensus on—or at least, machine-processable documentation of—the *common data elements* being recorded. There can be no magic here: if two trials have collected incompatible data, or one of them has provided insufficient metadata to allow compatibility to be determined, then their results cannot usefully be integrated. On the other hand, it is very difficult to arrange for prior universal agreement on compatible data elements across a large, heterogeneous, and long-lived community. The approach to this dilemma that we have taken on the CancerGrid project is realist rather than idealist. We have developed tools to support communities in deciding on, recording, and disseminating data standards; but there is no need for all parties to commit to using the same standard. Data elements (for example, 'blood pressure on induction

into study, measured at rest') are curated in the metadata registry, and referenced in the trial protocol, and the metadata reference is preserved in the software artifacts generated from the protocol—data entry forms, database schemas, spreadsheets, web services, and so on—ensuring that all the data maintain their semantic annotations throughout their journey through the system. We discuss this aspect of the approach in depth in Section 2.

Regarding model-driven engineering, an early activity of the CancerGrid project was the reification of the *Consolidated Standards of Reporting Trials* (CONSORT) statement [2] as a domain model of clinical trial design and execution. The scientist designing a trial follows the guidelines in the CONSORT statement to construct a model of the trial—we therefore describe the CONSORT domain model as a *metamodel* of clinical trials, and the design of an individual trial as a *model* of that trial. We have developed tools to generate the software artifacts needed to execute a trial from the model. We discuss our approach to model-driven engineering in Section 3.

2. Metadata-based data integration

2.1. Meta-analysis

As we have seen, meta-analysis is a crucial part of present-day medical research. For example, consider the drug Tamoxifen, used for the treatment of certain types of breast cancer. It was approved for use from 1980, but the available evidence from clinical trials for its efficacy was mixed throughout the 1980s, and it wasn't until a decade later that a meta-analysis [3] of the results from these trials, conducted by the Oxford-based Early Breast Cancer Trialists' Collaborative Group, produced useful results. As Richard Gray, a member of the EBCTCG secretariat, put it: "The drug Tamoxifen—an oestrogen blocker that may prevent breast cancer cells growing—was the object of forty-two studies world-wide, of which only four or five had shown significant benefits. But this did not mean that Tamoxifen did not protect against breast cancer. When we put all the studies together it was blindingly obvious that it does" [4]. This meta-analysis allowed researchers to identify the subset of the population responsive to the drug, and the optimum treatment regime. It yielded evidence that changed UK clinical practice, reducing mortality from operable breast cancer by 24%.

On the other hand, meta-analysis is no silver bullet: it doesn't always produce useful results, even when there is data from many trials to draw upon. Consider the case of the TP53 gene, which may be a useful prognostic marker allowing the prediction of outcomes to therapy for ovarian cancer in individual patients. One systematic review of 75 clinical studies, involving 8331 patients, could draw no useful conclusions: most of the study metadata was missing, and results were reported in insufficient detail to justify comparison. Even some less ambitious meta-analyses have stumbled over inconsistent study design and reporting: "it remains of the utmost importance to reach a consensus about guidelines for the design, conduct and analysis of such studies in ovarian cancer" [5], and "the data demonstrate the importance of methodological standardisation, particularly defining patient characteristics and survival end-point data, if biomarker data from multicentre studies are to be combined" [6].

2.2. Multiple perspectives

In an ideal world, perhaps, all researchers working in the same field would agree in advance all the details of the nature and the format of the data they will collect—the "guidelines for the design, conduct and analysis of studies" and "methodological standardization" referred to above, but also the data formats to be used, the storage media, communications protocols, consent criteria, and any number of similar stumbling blocks to data integration. But pragmatic concerns rule out such idealism: local considerations, historical accidents, political expediency, and personal preference all lead quite reasonably and rationally to differences in approach.

It is not even clear that complete uniformity is to be desired, anyway. The notion of 'the same field of research' is getting more inclusive all the time; connections arise between what were formerly seen as separate diseases, and current trends in translational medicine are bringing research from other branches of the physical and social sciences to bear on clinical practice; there will always be disciplinary borders to cross, and consequential procedural differences to reconcile. And besides, scientific understanding and best practice evolves, and guidelines that were appropriate for an earlier age may no longer be sufficient today; there will always be temporal borders between 'old' and 'new' too. Absolute uniformity is surely an unrealistic dream.

In fact, the more that science lifts its ambitions towards transdisciplinarity, towards internationalism, towards grand challenges, the more it has to accommodate divergent and sometimes inconsistent perspectives—the more it

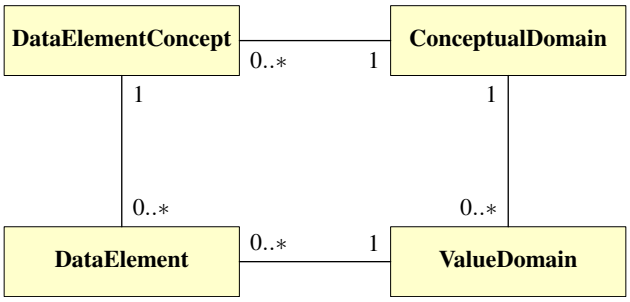

Figure 1: Associations between 11179 metadata element classes

has to abandon the simplicity of modernism, and embrace postmodernity. Simplifying to the extreme, postmodernity has been defined as "incredulity towards metanarratives" [7], the relinquishing of the expectation of a single universal framework for discussion. In particular, as the scope of our information systems extends beyond a single program run by a single person on a single machine to encompass broader collaborations across space, time, and discipline, we have to accept that "no requirements can be both complete and consistent: you have to pick one" [8]. This is the battle cry of the *postmodern programming* movement, which "rejects overarching grand narratives. As a result, it favours descriptive reasoning rather than prescriptive. Rather than working top down from a theory towards practice, postmodern programming theories are built up, following practice. Moreover, theory follows practice on a case-by-case basis" [8].

2.3. Metadata registries

To enable meta-analysis, then, scientific communities need to deal with metadata describing the design, execution, and analysis of the studies they conduct. A small, coherent, homogeneous community may find it helpful to produce a shared catalogue of standard metadata to draw upon: aspects of trial documentation (such as name, unique identifier, funding source, principal investigator); standardized clinical observations; reusable sections of forms (such as for subject identification); and so on. A community-wide shared catalogue of this form can be very helpful in promoting data standards, greatly facilitating data integration and meta-analysis. But when the community gets too large or too diverse to converge on such a shared catalogue, it is just as important to allow for local practice and special needs, maintained in separate metadata registries.

ISO/IEC 11179 [9] is a standard for metadata registries, envisaging a distributed network of metadata databases describing data assets in terms of 'metadata elements'. These metadata elements might be used retrospectively, to catalogue an existing dataset, or prospectively, in the construction of a data model for a new dataset. Part 3 (of 6) of the standard sets out a metamodel for metadata registries: a metadata registry is defined as a collection of metadata elements, which are subdivided into four types—*data elements*, *value domains*, *conceptual domains*, and *data element concepts*.

The metamodel is broadly split into a *representation layer*, which describes how observations and values in datasets are encountered, and a *conceptual layer*, which organizes content according to an overall notion of the domain of the registry. The metamodel declares a number of associations between metadata items: a data element shall be associated with exactly one value domain and one data element concept; a value domain could be declared in the absence of a data element, but can be shared by many, and is also associated with at least one conceptual domain (see Figure 1).

Roughly speaking, the left-hand column of Figure 1 shows 'questions', the right-hand column 'sets of valid answers'; the top row is the conceptual layer, and the bottom row the representation layer. For example, the data element 'body mass in kilograms, on entry into the study' might be associated with the value domain 'mass, to the nearest kilogram', the data element concept 'body mass', and the conceptual domain 'mass'; a second representation-level data element 'body mass in grams, after treatment' might be associated with value domain 'mass, in grams', and with the same two elements at the conceptual level. Different representation elements associated with the same conceptual element are intended to be commensurate: thus, whereas the value domain specifies the units of measure, the conceptual domain specifies only the dimensionality.

In providing a general metamodel for elements of data models, ICE/IEC 11179 occupies an important place in model-driven engineering systems: it offers an extensible source of reusable metadata that can be used to embellish UML, W3C XML Schema, and relational models, achieving more complete generation of system artifacts—particularly user interface elements, where choice of control groups, labels on forms, and column headings can be constrained or inferred from metadata recorded in the registry. Eventually ISO/IEC 11179 will be accompanied by standards describing the registration of models in the ISO/IEC 19763 series on metamodelling; however, these standards are at an early stage of development, and are not yet well integrated with ISO/IEC 11179.

To support recording and curating of metadata, standardized where possible but customized where necessary, we have developed the *CancerGrid Metadata Registry* (cgMDR) [10]. The cgMDR is robust enough for widespread use—it is currently being adopted by the US National Cancer Institute (`http://www.cagrid.org/display/MDR/`), for example—while still being lightweight enough for individual trials units to install their own copy to support local variations. It is the first free and open-source implementation of the ISO/IEC 11179 standard (NCI caBIG, the US National Cancer Institute Cancer Biomedical Informatics Grid, have an open-source ISO/IEC 11179 metadata repository caDSR [11], but it depends on having an Oracle license).

3. Model-driven software development

3.1. The CONSORT Statement

The CONSORT ('consolidated standards of reporting trials') statement [2] is intended to capture best practice in reporting randomized controlled clinical trials. It specifies a checklist of twenty two items, such as title and abstract, eligibility criteria, interventions, sample size, randomization protocol, adverse events, and so on. It also specifies a workflow of trial execution, depicting the passage of participants through the lifecycle of a trial—enrolment, intervention allocation, follow-up, and analysis. Retrospectively, CONSORT promotes understanding of a trial's design, conduct, analysis, and interpretation in order to assess its validity, through transparency on the part of trial authors. Prospectively, it also guides authors in how to set up a trial to maximize the utility of the results.

CONSORT has become a widely adopted standard in the reporting of clinical trials; for example, the International Committee of Medical Journal Editors' manuscript submission requirements [12] state that "Articles on clinical trials should contain abstracts that include the items that the CONSORT group has identified as essential", and the Nature group's policy on availability of data and materials [13] states that "Authors reporting phase II and phase III randomized controlled trials should refer to the CONSORT Statement for recommendations to facilitate the complete and transparent reporting of trial findings. Reports that do not conform to the CONSORT guidelines may need to be revised before formal review."

The CONSORT statement was originally published in 1996 [14]; it was revised in 2001 [15], and again in 2010 [16]. The CancerGrid trials metamodel was based on the 2001 version, the latest one at the time. However, the revisions have all been minor: it would be a simple exercise to revise the CancerGrid metamodel to match the current version of the statement. In the interests of historical accuracy, the discussion in this paper is based on the 2001 version.

3.2. Trials metamodel

The first step in any model-driven engineering endeavour is to establish a *domain metamodel*, circumscribing the area of applicability. We call it a 'metamodel', because its instances are models—each instance describes a particular application, such as a clinical trial, and forms the basis of the generation of software artifacts customized to support that particular application.

Scoping this metamodel appropriately is essential: too broad, and it becomes impossible to properly address the variance in an automatic manner; too narrow, and the toolchain is likely to be too concrete, and too dependent on specific instances, missing opportunities for wider application.

But just as important is the effort spent in ensuring that the metamodel accurately captures what is important to domain specialists. In that respect, CONSORT was extremely valuable to the CancerGrid project: experts in designing, conducting, and reporting clinical trials have devoted huge amounts of time and energy into characterizing best practice in their domain, explaining the rationale behind their modelling decisions [17], and obtaining buy-in and

	Item number	Descriptor
Title and abstract	1	*How participants were allocated to interventions (eg, "random allocation", "randomised", or "randomly assigned").*
Introduction		
Background	2	*Scientified background and explanation of rationale.*
Methods		
Participants	3	*Eligibility criteria for participants and the settings and locations where the data were collected.*
Interventions	4	*Precise details of the interventions intended for each group and how and when they were actually administered.*
Objectives	5	*Specific objectives and hypotheses.*
Outcomes	6	*Clearly defined primary and secondary outcome measures and, when applicable, any methods used to enhance the quality of measurements (eg, multiple observations, training of assessors, &c).*
Sample size	7	*How sample size was determined and, when applicable, explanation of any interim analyses and stopping rules.*
Randomisation		
Sequence generation	8	*Method used to generate the random allocation sequence, including details of any restriction (eg, blocking, stratification).*
Allocation concealment	9	*Method used to implement the random allocation sequence (eg, numbered containers or central telephone), clarifying whether the sequence was concealed until interventions were assigned.*
Implementation	10	*Who generated the allocation sequence, who enrolled participants, and who assigned participants to their groups.*
Blinding (masking)	11	*Whether or not participants, those administering the interventions, and those assessing the outcomes were aware of group assignment. If not, how the success of masking was assessed.*
Statistical methods	12	*Statistical methods used to compare groups for primary outcome(s); methods for additional analyses, such as subgroup analyses and adjusted analyses.*
Results		
Participant flow	13	*Flow of participants through each stage (a diagram is strongly recommended). Specifically, for each group, report the numbers of participants randomly assigned, receiving intended treatment, completing the study protocol, and analysed for the primary outcome. Describe protocol deviations from study as planned, together with reasons.*
Recruitment	14	*Dates defining the periods of recruitment and follow-up.*
Baseline data	15	*Baseline demographic and clinical characteristics of each group.*
Numbers analysed	16	*Number of participants (denominator) in each group included in each analysis and whether the analysis was by intention to treat. State the results in absolute numbers when feasible (eg, 10/20, not 50%).*
Outcomes and estimation	17	*For each primary and secondary outcome, a summary of results for each group, and the estimated effect size and its precision (eg, 95% CI).*
Ancillary analyses	18	*Address multiplicity by reporting any other analyses performed, including subgroup analyses and adjusted analyses, indicating those prespecified and those exploratory.*
Adverse events	19	*All important adverse events or side-effects in each intervention group.*
Discussion		
Interpretation	20	*Interpretation of the results, taking into account study hypotheses, sources of potential bias or imprecision and the dangers associated with multiplicity of analyses and outcomes.*
Generalisability	21	*Generalisability (external validity) of the trial findings.*
Overall evidence	22	*General interpretation of the results in the context of current evidence.*

Figure 2: Checklist of items to include when reporting a randomized trial [15]

support from their community. It was immediately clear that the CONSORT statement ought to form the basis of the CancerGrid domain metamodel.

Figure 2 shows the CONSORT checklist of items to include when reporting a randomized trial. We expressed this checklist as a UML class metamodel of well-designed trials. This metamodel is quite elaborate, but is partitioned into packages defining trial description, patient eligibility, randomization, treatments, case report forms (CRFs) and form controls, and security policy. A fragment of the metamodel representing form controls is shown in Figure 3. This aspect of the CancerGrid philosophy is described in more detail elsewhere [18].

In the terminology of the Object Management Group's *Model-Driven Architecture* [19], a particular clinical trial is a real-world system, at level M0; the trial protocol is a model of that system, at level M1; our UML rendition of the CONSORT statement is a domain metamodel, at level M2; and the metametamodelling framework at level M3 is not formalized.

3.3. Trial modelling

In the next step, a scientist planning a clinical trial designs the trial protocol—in terms of aspects such as eligibility criteria and interventions covered by the CONSORT statement, together with basic 'Dublin Core' metadata items such as title, author, and date. Traditionally, they would do this by writing a prose document describing the experiment they intend to conduct, following the guidelines in the CONSORT statement. This document might form the basis of an application to an ethics committee for permission to conduct the trial, and of a proposal to a funding body to obtain the resources needed to support the trial.

But given a rigorous metamodel, it is possible to do better than this. It is straightforward to automatically export the domain metamodel to yield a data schema for domain models; then the scientist would describe their planned experiment by producing a data object—a model of the trial—that conforms to the data schema. This experimental model will necessarily conform to the CONSORT guidelines, which cannot be said for the prose document written in the traditional way.

Depending on the notation in which the metamodel is expressed, the scientist can be given considerable assistance in producing their trial model. In Section 4.1 below, we present a worked example in which the metamodel is exported as an XML Schema, and the 'trial designer' application that the scientist uses is simply Microsoft InfoPath configured with the metamodel. Designing a trial amounts to completing the forms that InfoPath presents. The result is an XML document modelling this particular trial that, by construction, conforms to the schema and to the CONSORT statement.

3.4. Model-driven generation

Once the scientist has produced a model of the trial that they intend to conduct, in a suitably structured notation, it is possible to generate from that model a collection of software artifacts to support the trial. These include forms for data collection, web services for validating and storing completed forms and for randomizing allocation of patients to treatment arms, blank spreadsheets for analysing data, textual documents for reporting, workflow monitors for guiding trial execution, and so on.

Mostly, these software artifacts are themselves just various forms of structured data, and are readily generated by transforming the trial model. For example, blank forms, reports, and workflow models might be represented in dialects of XML. Services don't directly fit this pattern; for these, one can write a generic service (for example, for randomization) once and for all, and generate a trial-specific configuration file automatically from each trial model. Some concrete examples are given in the case studies presented in Section 4.

3.5. Trial execution

The most important of the software artifacts needed to conduct a trial turns out to be the simplest to generate: the *case report forms* that are completed as the trial progresses, recording patient information and clinical observations. A natural representation for a blank form is in terms of the data schema for the data that form is designed to collect; that might be written in XML Schema, which is of course a structured data format, and readily generated from the trial model. Completing a case report form therefore amounts to constructing a structured data object (perhaps an XML document) conforming to the data schema for that form. Again, this process may be managed by a standard forms completion application.

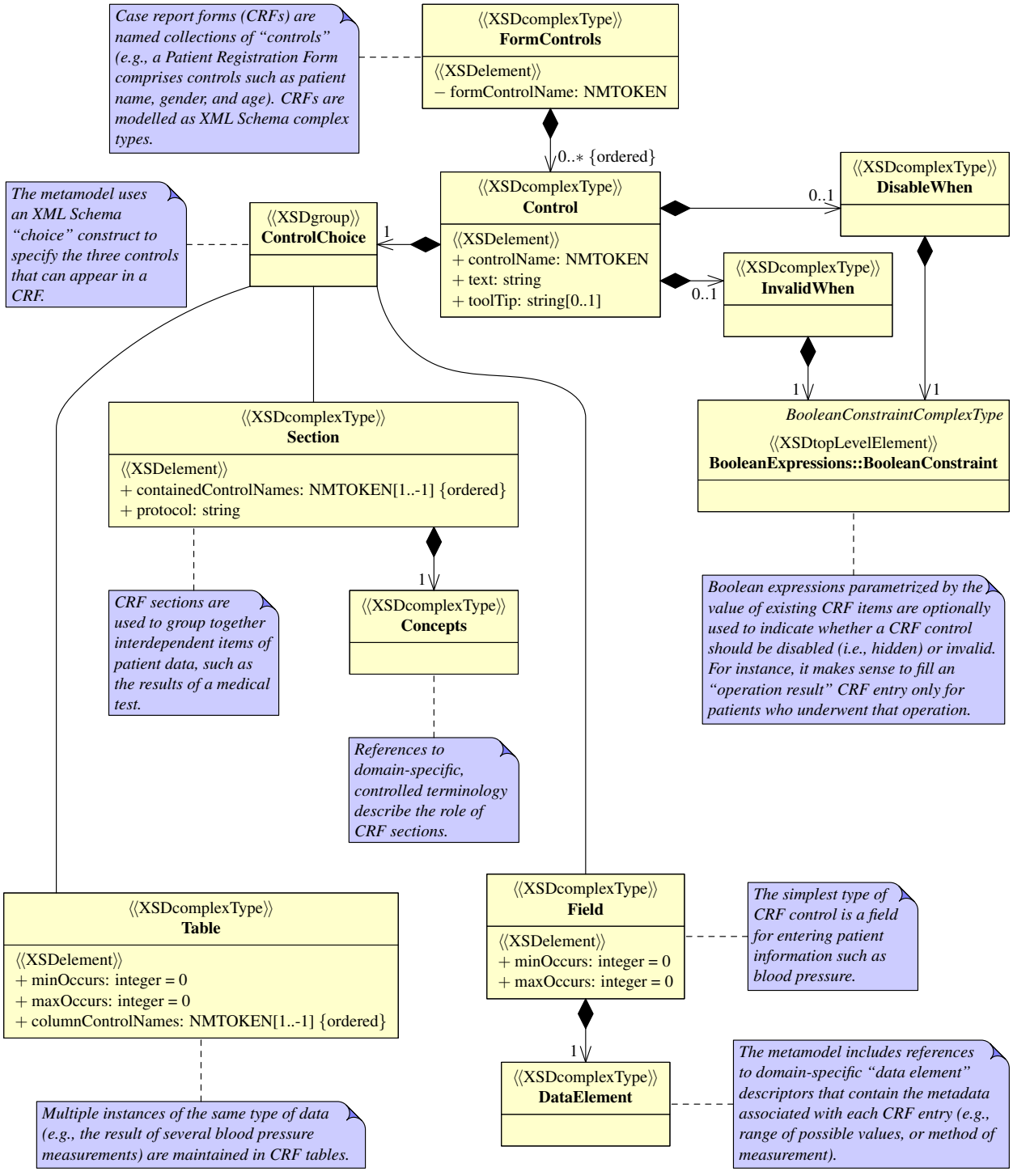

Figure 3: UML class model of the Form Control package of the CancerGrid metamodel

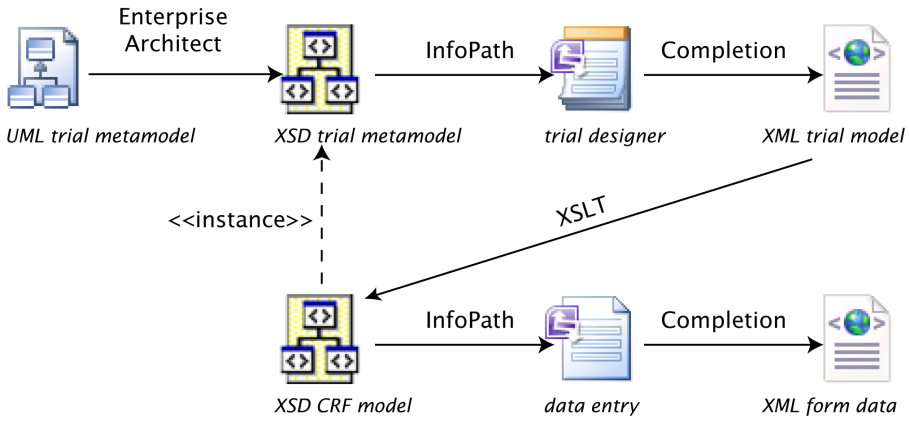

Figure 4: The forms-based workflow of clinical trial design and execution; icons denote artifacts, and solid arrows denote transformations between artifacts

An intriguing aspect of the CancerGrid approach is therefore the parallels it reveals between the 'design-time' and 'run-time' stages in the lifecycle of a trial. In the first stage, a scientist is designing a trial protocol, following the guidelines set out in the CONSORT statement; in the second stage, a clinician is conducting a trial, following the guidelines set out in the trial protocol. Both cases involve instantiating a schema: the trial protocol instantiates the CONSORT class model, but it determines schemas for data collection and reporting at significant events during the trial, which are themselves instantiated by the clinician when entering data.

4. Case studies

4.1. Prospective data collection

As the name suggests, the original exemplars for the CancerGrid project were in cancer clinical studies, and in particular in breast cancer trials [20, 21, 22]. But of course, the approach is not limited to cancer; a more recent application has been a project with the Oxford Vaccines Group to study the efficacy of a paediatric pneumococcal vaccine in Kathmandu [23]. We walk through that application here, in order to illustrate in detail the CancerGrid approach to data collection.

As discussed in Section 3.3, when the domain metamodel is sufficiently simple and regular, constructing a domain model can be as simple as completing a form. This observation is explored in depth in a recent paper [24]; we summarize that presentation here. The general principle is one of data models as document schemas, entities as conformant documents, authoring as form completion, and model transformations as schema mappings. This principle can be realized in a variety of ways. Our current implementation uses an off-the-shelf product for form-filling—namely Microsoft InfoPath, an application that supports the design and completion of XML-based data entry forms, forming part of the Microsoft Office productivity suite. This has turned out to be the approach most attractive to our target audience of medical researchers and clinicians, because they are all familiar with the Microsoft Office interface, and they all have the software pre-installed on their desktop computers. However, the general principles are in no way tied to this particular realization, and our earlier paper [24] also presents an alternative implementation developed from first principles.

The workflow entailed by trial design and execution is illustrated in Figure 4. The preliminary step, in which representatives of the data community develop a domain metamodel, is discussed in Section 3.2 above; this is shown in the figure as a UML metamodel, which is exported from the UML modelling tool (for example, Enterprise Architect) into an XML Schema representation.

First, the medical researcher planning a trial specifies the trial protocol, using a 'trial designer' application. In this case, the trial protocol is sufficiently formulaic to be specified by completing a form, and the 'trial designer' application is simply a generic form-filling application (such as Microsoft InfoPath) configured with the XML Schema representation of the trial metamodel. Figure 5 shows a screenshot of InfoPath being used to design a case report form

for recording study participant registration; the highlighted element is the 'study group' (a three-way enumeration) into which the participant is placed, and a version of the underlying XML representation of this piece of the form, heavily edited for space and readability, is shown in Figure 6(a).

The XML document recording the trial protocol determines numerous software artifacts relating to the trial: clinical interventions, datasets and collection procedures, software configurations for services such as randomization and validation, documentation, and so on. Each trial-specific artifact essentially instantiates the template for such artifacts included in the metamodel—hence the dashed realization arrow in Figure 4 stereotyped '⟪instance⟫', in a slight abuse of notation. (To avoid clutter, the only trial-specific artifact shown in the figure is the model of a case report form. Other artifacts are also models, but may be models of entities such as services, documents, or workflows, rather than of forms.) The specification of each artifact is obtained by traversing the XML document, extracting the corresponding parts, and transforming them into the appropriate format: XML Schema, XSL Formatting Objects, WSDL, and so on. Traversal, extraction, and transformation is specified as a collection of XSLT stylesheets, written once only, for all trials based on the same version of the metamodel. In particular, the data to be collected by the clinician conducting the trial—and hence the structure of the form on which this data is recorded—is specified by an XML schema. In the second step, the data manager in the trials unit generates all these artifacts automatically, and deploys them in the unit's web portal for access by clinicians. Continuing the example, the XML defining the participant study group is used to generate an XML Schema for the data that should be collected; the corresponding portion of that schema, again heavily edited for readability, is shown in Figure 6(b).

Finally, the clinician in the unit running the trial conducts a consultation with a participant in the trial, makes some clinical observations, and needs to record the data so obtained. The *data entry* application that they use to do so is just InfoPath configured with the model of the relevant data. Data entry amounts to completing the form that InfoPath presents; the result is an XML document recording this event that, by construction, conforms to the appropriate schema from the previous step, and which may be stored in an XML database for subsequent analysis and study. In the example, choices for the participant study group are presented as a dropdown list, as shown in Figure 7; and selection from this list results in the creation of the XML in Figure 6(c).

4.2. Retrospective data integration

One retrospective data analysis application of the CancerGrid approach was undertaken in the context of the Molecular Taxonomy of Breast Cancer Internal Consortium (METABRIC) [25], a collaborative study involving five hospitals in the UK and Canada, using molecular profiling in order to understand the clinical heterogeneity of breast cancers. The hospitals in the consortium all use different data definitions; they often hold incomplete datasets, because of patient movements; and record keeping and clinical procedures are likely to have changed over time during the relatively long treatment periods. CancerGrid metadata technologies helped the researchers to accommodate this variety in a straightforward and lightweight way.

To perform any kind of integration of heterogeneous datasets, some kind of data conversion will be necessary. For example, in the METABRIC study, each of the five hospitals used a different enumeration as a classification scheme for 'histological tumour type', ranging from 15 to 33 different codes; similarly, in some hospitals the data element 'menopausal status at diagnosis' was an enumeration, explicitly and directly determined by a clinician, but in others it was simply a boolean, inferred indirectly from whether the 'age at diagnosis' was at least 50 years. Such differences need to be reconciled before the data can be integrated.

Of course, data transformations of this kind typically lose information; distinct classification schemes are unlikely to match up precisely, and missing data might have to be estimated based on the data present. Crucially, what constitutes acceptable information loss will differ from integration exercise to integration exercise. It is a fool's errand to search for a grand unifying data reconciliation; it is much better to allow each specific data integration exercise to specify appropriate data mappings. Consequently, it is important that the data mappings are simple to specify and to implement—otherwise data integration becomes too laborious. Indeed, the data mappings should be modelled in a lightweight way, and their implementations should be generated from these models.

In the METABRIC project, we supported the medics, clinicians, and pathologists from each hospital in using the CancerGrid Metadata Repository in curating the data elements describing each field in their own data, and in each of the fields of the minimum dataset common to the whole consortium; there were 50 and 29 of these, respectively. We interacted with pathologists from each of the five institutions to categorize the 50 hospital-specific data elements into

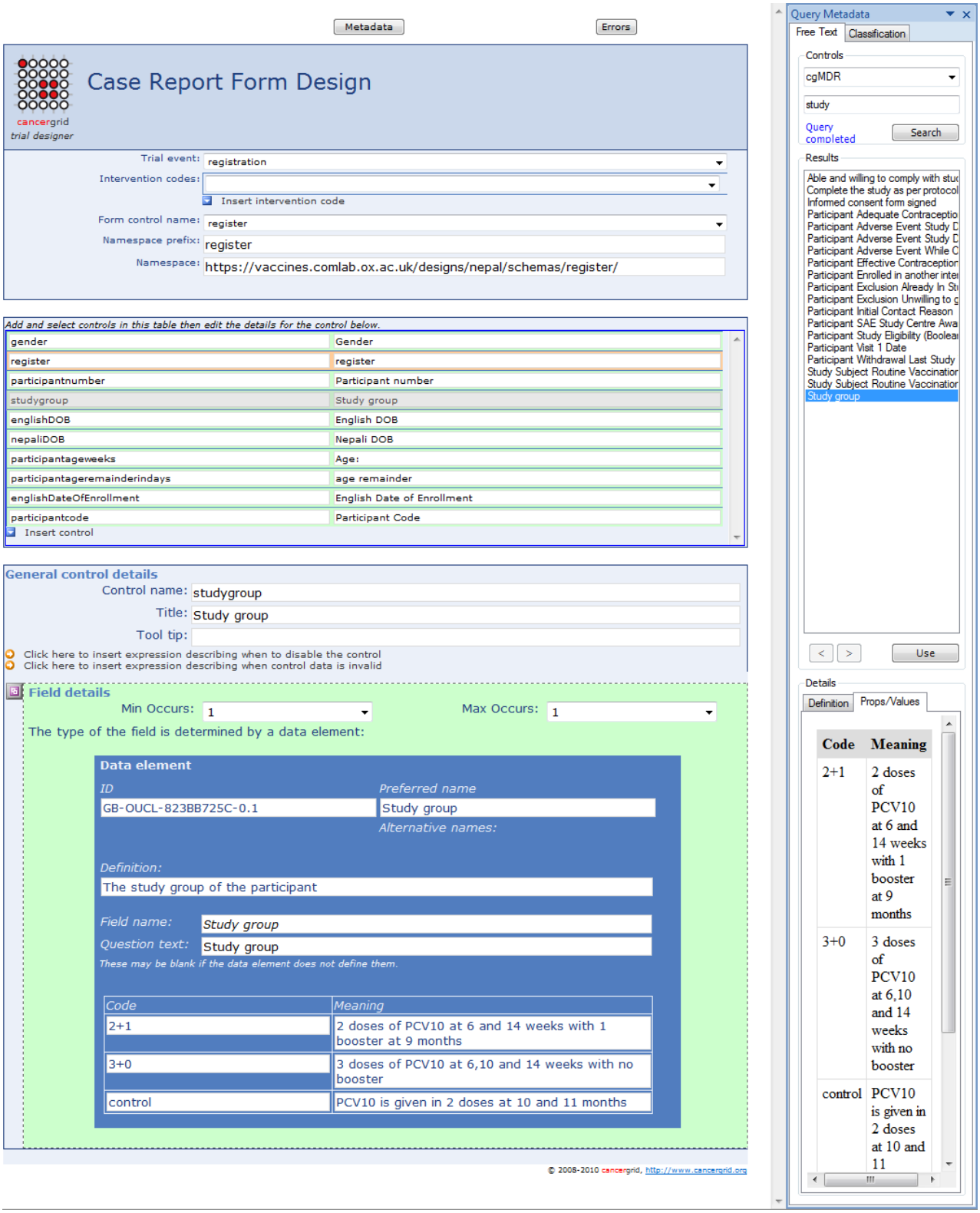

Figure 5: Using InfoPath to design a case report form

(a)

```
<controlName>studygroup</controlName>
<text>Study group</text>
<minOccurs>1</minOccurs>
<maxOccurs>1</maxOccurs>
<data-element>
  <id>GB-OUCL-823BB725C-0.1</id>
  <definition>The study group of the participant</definition>
  <valid-value>
    <code>2+1</code>
    <meaning>2 doses of PCV10 at 6 and 14 weeks with 1 booster at 9 months</meaning>
  </valid-value>
  <valid-value> ... </valid-value>
</data-element>
```

(b)

```
<xs:element name="studygroup" type="register:SimpleType_studygroup"/>
<xs:simpleType name="SimpleType_studygroup"
               sawsdl:modelReference="GB-OUCL-823BB725C-0.1">
  <xs:annotation><xs:documentation source="definition">
    The study group of the participant
  </xs:documentation></xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="2+1" />
    <xs:enumeration value="3+0" />
    <xs:enumeration value="control" />
  </xs:restriction>
</xs:simpleType>
```

(c)

```
<register:studygroup>3+0</register:studygroup>
```

Figure 6: Fragments of XML representing (a) a trial model, (b) a data schema derived from the model, and (c) recorded results conforming to the schema

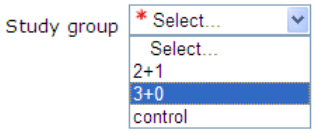

Figure 7: Form control for data collection

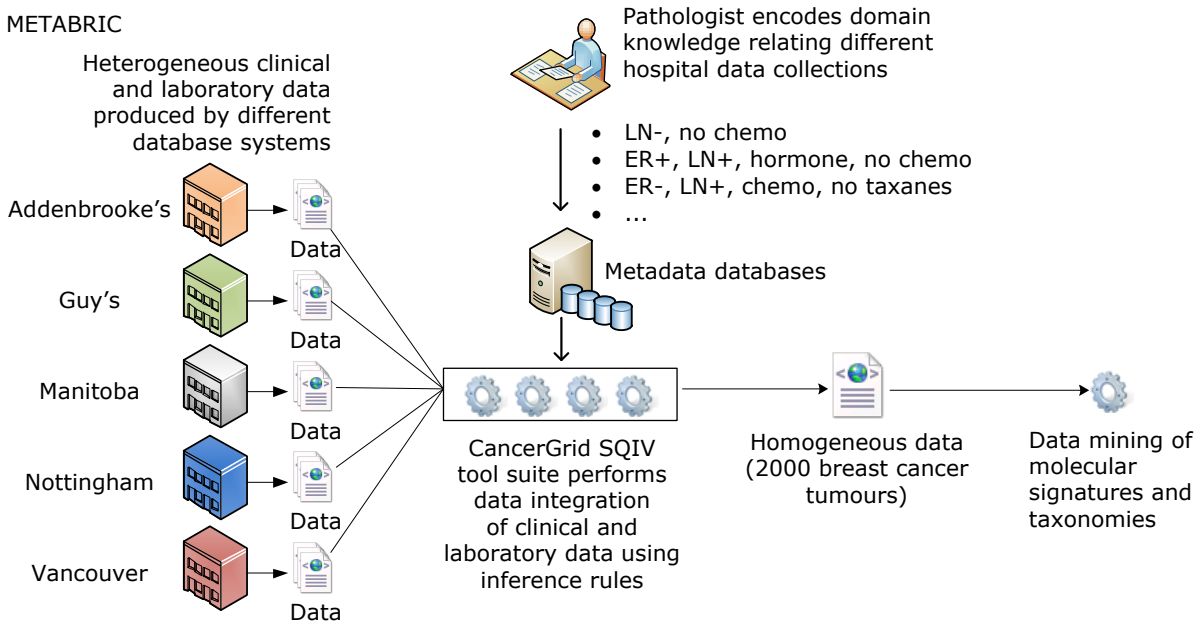

Figure 8: Using SQIV to annotate heterogeneous METABRIC datasets for meta-analysis

33 data element concepts; for example, the five hospital-specific data elements 'Addenbrookes histological tumour type', 'Guy's histological tumour type', and so on all correspond to the data element concept 'histological tumour type'. We then used SQIV [26]—a home-grown suite of tools supporting the standardization, querying, inference upon and validation of data corresponding to XML schemas marked up using SAWSDL—to annotate the datasets with the common categorizations. Figure 8 illustrates the SQIV process. The annotation was performed by rule-based inference based on Horn clauses, using the Jena semantic web framework [27]. This reconciliation enabled a meta-analysis over the combined and somewhat heterogeneous datasets, amounting to about 4000 samples in total, which would have been impractical under existing manual processes.

It would be entirely straightforward to generate the relevant Horn clauses automatically from the metadata in the MDR; tools such as ATLAS Model Weaver [28] for establishing links between models are very helpful in this kind of exercise, because they allow the domain specialist designing each individual data integration task to say simply, directly, and precisely which values may be considered 'the same' for their particular purposes. (As it happens, however, we constructed the Horn clauses by hand, since only a handful of them were needed for this exercise.)

4.3. Data cataloguing

In clinical trials, it is important to ensure that negative results do not get buried, and to prevent experiments from being repeated in private until the desired outcome has been achieved. To address this concern, prospective study registers are of paramount importance [29, 30]. In 2005, the International Committee of Medical Journal Editors (ICMJE) made prospective registration a prerequisite for publication in any of its member journals; in the following month, the number of trials registered with `clinicaltrials.gov` nearly doubled [31].

Traditionally, experimental registers have offered a uniform, one-size-fits-all model for the type of experiment registered, and this has been a considerable success. However, with half a million records in the Cochrane Central Registry of Controlled Trials [32], simple browsing or Google-style search is becoming more difficult, and it becomes pressing to be able to obtain more detailed information about the execution of the study in order to further filter results, even before finer considerations of data compatibility are made. As one wishes to record more detail about a study, each study becomes increasingly distinct, and one needs to admit subtypes of studies to be able to properly record metadata about them. Despite requirements for central registration, there are still a considerable number of disease-specific registries in response to these problems. Ideally one would want to see interoperation between disease-specific

registries, and facilities for the easy interchange of records. We believe the best way to support these requirements is in the development of generic registry software that can be customized by extension to the registry model.

In the related area of longitudinal population studies, we have been able to use model-driven techniques to provide a toolbox for the creation of a specific ISO/IEC 11179 metadata registry; a screenshot is shown in Figure 9. Unlike phase III clinical trials, which support the evaluation of a single, testable hypothesis through a clear and often simple statistical analysis, a longitudinal population study establishes a large, interesting cohort of many thousands of people, which is recalled for a wide variety of purposes over decades, integrating clinical, psychological and social evaluations with public records such as those collected in schools and healthcare systems. A typical population study in the UK is the Avon Longitudinal Study of Parents and Children (ALSPAC, or 'Children of the 90s') [33], which established a cohort of over 10,000 children with their parents, and was designed to determine how an individual's genotype combines with environmental pressures to influence health and development. Data gathered from ALSPAC supports a wide variety of studies: in the first few months of 2012, papers were published on the effects of dog ownership on obesity in pregnancy [34], bullying and suicide-related behaviour in 11-year-olds [35], and patterns of alcohol abuse in adolescence [36], all based on the ALSPAC study.

Using a basic implementation of ISO/IEC 11179 it is a relatively direct task to register and document the definitions of the many variables collected and curated in a study (nearly 40,000 in ALSPAC), recording the name of the variable, a brief description, and the value set or the applicable units. By taking the ISO/IEC 11179 notion of an 'administered item'—an item of metadata for which we wish to maintain provenance and exercise version control—we have been able to model appropriate content types in UML and generate XML Schemas, XForms, specific reference document types, XML Database Collections and menu-items to provide much of the functionality of the registry without manual coding. This facilitates the deployment of 11179-like metadata registries, whilst ensuring compliance to the core standard.

In the UK Medical Research Council Data Support Service pilot, through which we were working with the ALSPAC and other population health studies, the registration of metadata for population studies required the creation of new types of content: an overall study record containing textual information about study aims, cohort, status, recruitment, contacts, funding and data sharing policies; a 'dataset record' that allowed the naming and identification of sharable datasets, each reporting a particular cross-sectional analysis of the cohort for some purpose; a timeline record that could be used to associate particular metadata items with an appropriate Simile timeline (shown in Figure 10, for a different study in the DSS pilot); and a form model record that could better describe the association of variables collected in a single paper form, where the shared datasets did not correspond directly to individual form instances. Each content type was modeled in UML or XML Schema, and ANT scripts were used to generate successive customizations of the core registry to support the specified functionality.

4.4. Metamodelling forms

Evidently, forms are central to structured data collection procedures: not only do they constitute the user interface to a system, they also hugely influence the quality of the data collected, and inform its subsequent analysis. Consideration of the nature, design, generation, and use of forms has formed a significant part of the CancerGrid work. In this section, we summarize our work on modelling and metamodelling forms [37].

Three aspects of data quality that are particularly relevant to form design are correctness (the extent to which the values entered correspond to the intended interpretation), completeness (the extent to which the form faithfully accommodates the full story), and comprehensibility (the extent to which form data comes with adequate documentary metadata). Correctness depends on the intended interpretation and possible values of each data item being transparent, so that the person completing the form can work out how to respond. Completeness depends on a clear structure, helping the form completer to navigate between sections, pre-populating fields with default or inferred values, and hiding irrelevant questions and inappropriate answers. And comprehension of the results requires the form to contain links to appropriate metadata describing the terminology and the context.

We are therefore working on a domain-specific metamodel of forms; the position paper [37] presents progress to date. The intention is to enforce a clear separation of the concerns of internal structure, user presentation, content validation, and semantic annotation; and to do so in a compositional way, so that forms can be assembled from a library of form sections. The usual benefits of modelling apply: separation of concerns allows simpler and higher-level specification of what is required; it also supports multiple uses, such as documentation of a dataset, and advance

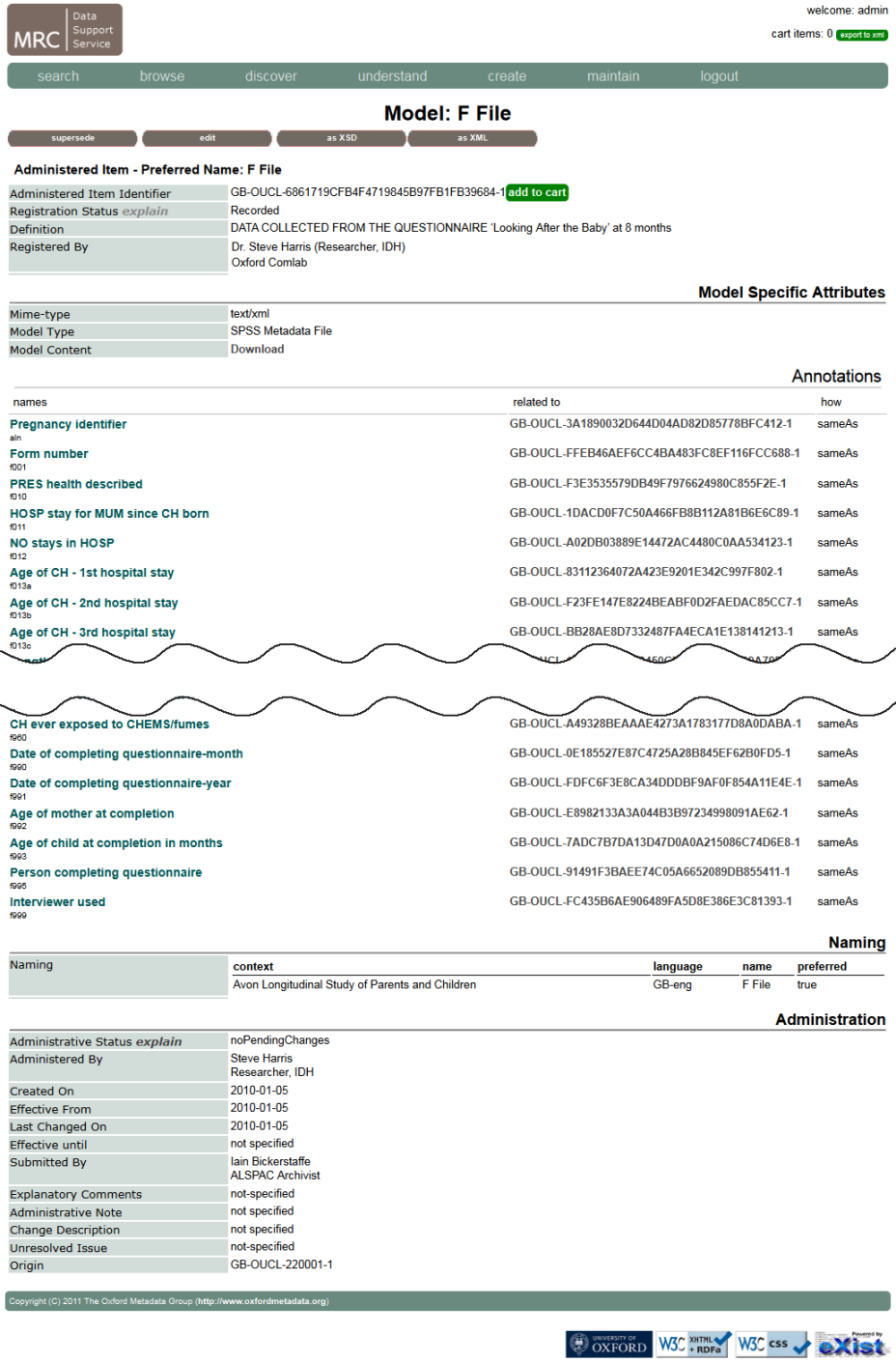

Figure 9: Screenshot of the DSS metadata registry

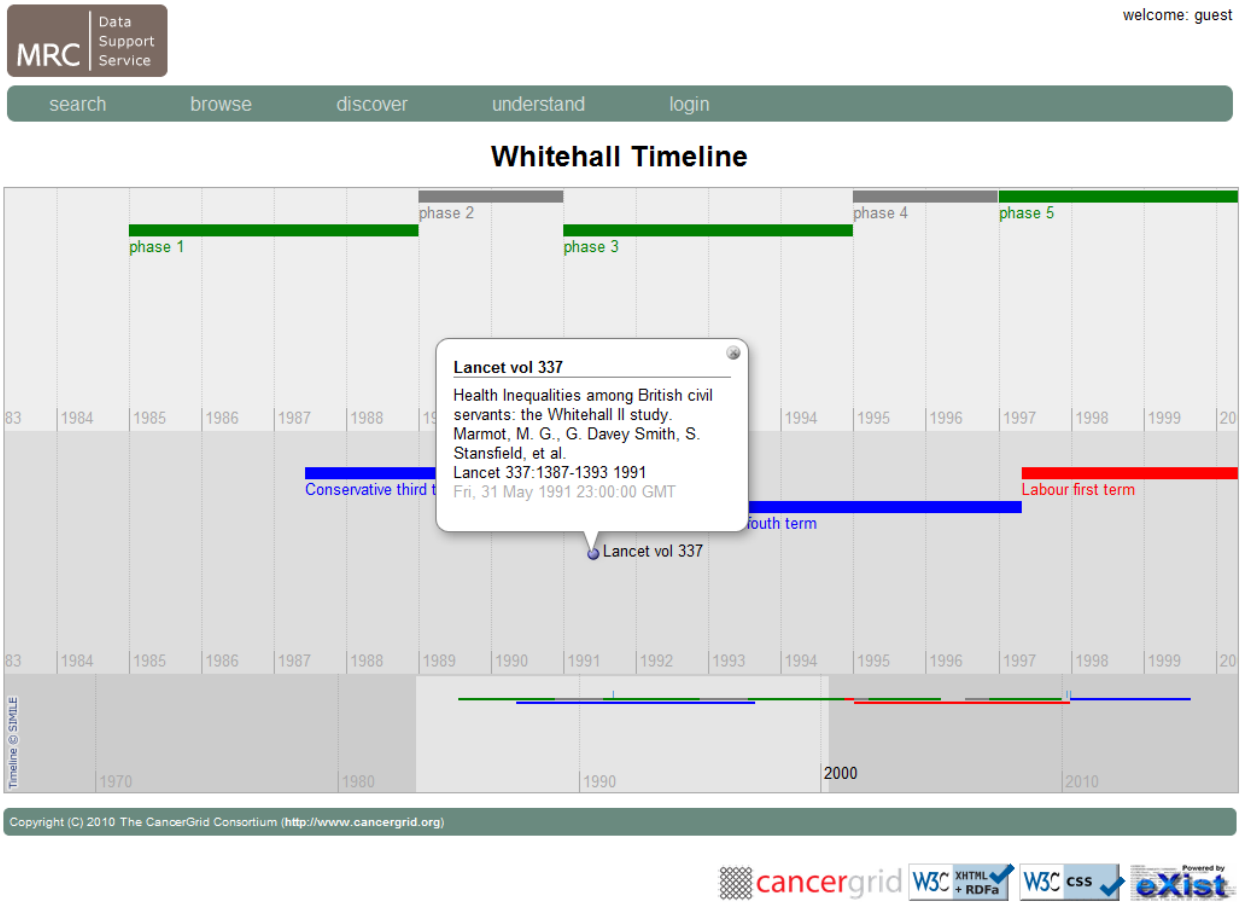

Figure 10: Screenshot of the Simile timeline for the Whitehall study in DSS

ITEM_NAME	LEFT_ITEM_TEXT	SECTION_LABEL	SUBHEADER	QUESTI(	RESPONSE_	RESPONSE_OPTIONS_TEXT	RESPONSE_VALUES_OI	DATA_TYPE
A_1a_DI_InformedConser	Was informed consent obtained for OCCAMS (which incorporates ICGC)	Demographics and Identifiers	Informed consent	DI.14	single-select	Please select, Yes, No, Do not know	,yes,no,do not know	ST
A_1a_DI_DateOfInformed	Date of informed consent	Demographics and Identifiers		DI.15	text			DATE
A_1a_DI_DateBloodsObta	Date bloods obtained	Demographics and Identifiers		DI.16	text			DATE
A_1a_DI_DateBloodsSent	Date bloods sent	Demographics and Identifiers		DI.17	text			DATE

Figure 11: Specifying an OpenClinica form via a spreadsheet

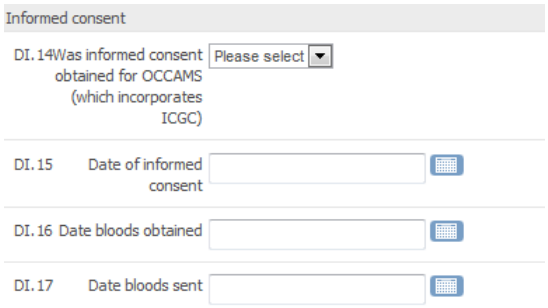

Figure 12: Data entry in an OpenClinica form

planning and coordination of data collection. Because forms are so widespread, this is an attractive domain in which to work—even modest results offer the promise of significant returns.

Naturally, there have been numerous previous attempts to improve the quality of data capture by formalizing the process. These attempts broadly fall into two classes: data documentation standards, such as the Data Documentation Initiative (DDI) [38] and the Clinical Data Interchange Standards Consortium Operational Data Model (CDISC ODM) [39], and common data capture platforms, such as OpenClinica [40] and REDCap [41].

Neither of these approaches have produced entirely satisfactory results. The data standards activity has focussed primarily on post hoc documentation: models of forms are used to record form contents and structure, but not to generate forms for data capture. Consequently, form modelling represents an additional burden on researchers, who may derive no tangible benefit themselves in terms of increased data quality and simpler reuse. Moreover, unless the models are generated automatically from the data, or vice versa, it is very difficult to ensure their consistency.

Conversely, although study management systems such as OpenClinica and REDCap may use form models as the basis of form generation and deployment, these models are relatively simplistic, but nevertheless rather difficult for non-technical domain specialists to understand (we explore an example below). Moreover, the models are not presented as documentation for the data captured, and are difficult to reuse outside the context of the particular study management system.

Neither the data documentation standards approach nor that of study platforms has resulted in a standard means of recording the logical relationships between questions or question sets asked in different forms or in different studies, except by reference to a shared data standard, data dictionary, or question bank. To determine whether observations from two different studies are comparable for the purposes of meta-analysis, it is neither necessary nor sufficient that they are based on the same standard question: insufficient, because the additional context provided by the form may change the interpretation of the question, and unnecessary, because the semantic imports of two syntactically quite different questions might nevertheless be compatible.

We have been investigating the use of OpenClinica in supporting a study of esophageal cancer as part of the International Cancer Genome Consortium [42]. Although OpenClinica does have a forms metamodel, the mechanism for modelling forms is rather obscure, at least to clinical researchers. Researchers provide input encoded in a spreadsheet, with entries in different columns representing question text, answer range, value constraints, and navigation rules; Figure 11 shows a fragment of the spreadsheet specifying a form for the ICGC study, and Figure 12 a screenshot of the corresponding part of the data entry window that OpenClinica generates. REDCap uses a similar mechanism [41].

The problem we found is that domain specialists cannot understand the spreadsheet modelling format, and so they are cut out of the loop when it comes to designing and commenting on a new form: many more people need to be

DI Demographics and Identifiers Version: PILOT_6

Informed consent

DI.14 Was informed consent obtained for OCCAMS (which incorporates ICGC)?

- ○ Yes
- ○ No
- ○ Do not know

DI.15 Date of informed consent: …… / …… / …… (d/m/y)

DI.16 Date bloods obtained: …… / …… / …… (d/m/y)

DI.17 Date bloods sent: …… / …… / …… (d/m/y)

Figure 13: Textual presentation of an OpenClinica form

consulted about the questions than ever need to actually use the system to enter data. To get around this problem, we produced a tool to generate a Word document from the spreadsheet form model; an example is shown in Figure 13. This has greatly improved the turnaround speed for discussing and modifying the study, as the domain specialists working on the study find reading and commenting on Word documents easier than learning how to interpret the spreadsheets or logging into the system to see the forms online. Currently, the tool is simply an ad hoc translation implemented in C# and XSLT; but the plan is to use the forms model discussed in the position paper [37] as a lingua franca, supporting translation to and from OpenClinica, Word, and InfoPath as used elsewhere in CancerGrid.

5. Conclusions

5.1. Summary

The CancerGrid project has taken a two-pronged approach to the development of software to support clinical trials. On the one hand, rich semantic metadata is a necessary prerequisite for subsequent data integration; experiment designers need to be able to record their intentions, and data collectors their actions, in such a way that later users can properly interpret the data. On the other hand, clinical trial management is ripe for a model-driven approach: the domain is rather formulaic, with the software development following well-trodden paths, so there are good prospects for automation; and it is common for busy experts in a wide variety of disciplines to need to be involved in design, encouraging high-level modelling rather than low-level programming. Happily, these two aspects reinforce each other: automation in the development of software tools makes it straightforward to propagate metadata alongside the data, preserving semantic annotations throughout analysis; and semantic information in the model can inform and refine the consequent generation of artifacts.

5.2. Experience

We have had seven years' experience with the approach, in a number of different clinical contexts with various different sets of clinical collaborators. The original CancerGrid project ran for three years from 2005. Subsequent funded projects applying the CancerGrid approach include: *Accelerating Cancer Research Using Semantics-Driven Technology* [43], funded by Microsoft Research, exploring the extension from phase III to early-phase studies; *Evolving Health Informatics* [44], funded by Research Councils UK, working with colleagues in the Centre for Clinical Vaccinology and Tropical Medicine at the University of Oxford to demonstrate applicability to infectious disease control; *Hospital of the Future*, aiming to improve patient outcomes through information-driven management; the *Data Support Service*, funded by the UK Medical Research Council (MRC), to retrospectively catalogue the data collected in some of the MRC's valuable long-running studies; and the *Union of Light-Ion Centres in Europe*, funded by the European Union Seventh Framework Programme, to curate experimental results in particle therapy. In addition, we have been collaborating with local colleagues on a pro bono basis. The vaccinology study [23] discussed in Section 4.1

is one such; we were able to reduce the time taken to produce a complete set of semantically annotated forms for the study from six months to as many weeks—not least because a set of prototype forms could be shown to clinical researchers with a turnaround of a few days, while the discussion of their design was still fresh.

5.3. Lessons learned

As the name of the project suggests, the original plan in 2005 was to use the then-emerging 'grid' technologies as a basis for the implementation. This turned out to be impractical: the toolkits were relatively unsophisticated, requiring considerable programming effort to duplicate the functionality of applications that were already available to our target users; moreover, 'grid computing' in the sense of large-scale computational or storage clusters is not relevant to this particular problem. The development activity was thus refocussed upon the production of software that worked to extend and configure applications that are widely used and available within the UK National Health Service (and, indeed, throughout government and industry): specifically, Microsoft Office and SharePoint Server. The project's focus upon the requirements of clinical researchers, and the recognition that these requirements can be met partly through the (automated) enhancement and configuration of office productivity applications, has led to changes in attitudes. As one team member put it: "We used to be hung up on open source; now we're really focussed on open standards."

The call for papers for this special issue observes that "Historically model transformations and code generation from abstract models have been among the main MDE applications. Nevertheless, they represent only a partial constituent of the MDE application ambit [...] applications emerging from the most disparate domains may reveal new directions for development of the theory as well as lessons of transferable value for future MDE practice." That observation is supported by our experience in the CancerGrid work. Model transformations are a relatively small part of the story, and code generation even smaller; much of the gain is obtained simply from the consistency and agility that are consequences of automation, and none of it would be of much use without consideration of other aspects such as appropriate semantic annotation.

We have encountered a number of non-technical obstacles to the model-driven approach we espouse. For example, we have had little traction in supporting the dynamic aspects of trial execution by exploiting a workflow model in the trial protocol. This is partly because it is not a particular pain point for the communities we are working with—current practice in randomized allocation of patients to treatment arms is for a statistician metaphorically to produce in advance a sequence of decisions in envelopes, which is a sufficiently low-technology practice to work well even in the most rudimentary of clinical contexts. Similarly, trials units generally already have well-established processes for exporting clinical data into statistical packages for subsequent analysis, and there was no obvious gain from integrating this aspect with the rest of the model-driven chain. Objections are also partly due to politics, especially an investment—individual or institutional—in more manual development techniques.

5.4. Applicability

It has been gratifying to see that the ideas we have developed are much more widely applicable than originally envisaged. Of course, we expected that software engineering efforts related to breast cancer ought to be readily translatable to other types of cancer, and we hoped that its area of applicability would also embrace other diseases; our results in vaccinology (and also in rheumatoid arthritis, in an exploratory collaboration with the US Veterans' Health Administration Cooperative Studies Program) have vindicated that hope. But we have been pleasantly surprised to learn that essentially the same approach is also highly relevant in the field of electronic government. This too turns out to be largely a problem of data integration: the former UK Prime Minister Tony Blair coined the term 'joined-up government' as a vision for how different government departments ought to—but generally do not at present—interact [45]. Moreover, electronic governance is also a domain in which model-driven generation of software artifacts would be extremely helpful: accountability of public servants requires government information systems to be transparent, and the monopoly typically held by the incumbent government requires the systems to be trustworthy. Our ideas in this area are still under development, but we have some preliminary results [46, 47, 48], and we are discussing further progress with the UK Public Sector Object Model group and with the Scottish Government.

Broadly speaking, we expect the approach to be applicable to any semantically rich domain in which there is: a relatively stable metamodel of the domain; a 'design phase' consisting of instantiating the metamodel to yield a model of a particular instance, which can be used to configure generic software tools; and an 'execution phase' in

which entities conforming to the model are created. We have been using the term 'semantic frameworks' to describe the approach when applied outside cancer clinical informatics [46]. For example, one could use the approach for a generic conference management system. The basis is a metamodel of academic conferences. The conference chair 'designs' the particular conference by instantiating the metamodel, specifying properties such as whether there is an author response period, whether reviews are double-blinded, how many reviewers each paper should have, and so on. 'Execution' consists of creating entities such as 'submissions' and 'reviews' that conform to conference-specific aspects of the model. The reader can doubtless think of many similar configurable information-gathering exercises.

5.5. Future work

One aspect of ongoing work is to extend the scope of the metamodel to cover also the temporal aspects of a clinical trial. Although the trial protocol provides structured specifications of static aspects, in terms of common data elements, the dynamic aspects—when interventions should occur—are described only in free text (see the 'meaning' fields in Figure 5). These too could be specified in a structured format in the protocol, in a workflow modelling notation such as BPEL or BPMN, and then used to generate scheduling tools for trial execution. We have conducted some preliminary studies on using such workflow notations to specify and check trial safety properties such as drug interactions [49, 50], but have not yet integrated this work with the rest of the CancerGrid toolchain. The biggest challenge will be to allow the trial designer to describe the temporal aspects of the trial in sufficient detail, without degenerating into a full-blown programming exercise; we hope that *workflow patterns* [51] and *property specification patterns* [52] will be helpful in this regard. But we are conscious that even if this can be made to work smoothly from a technical perspective, the human and organizational perspectives may still throw up obstacles to its adoption.

6. Acknowledgements

We would like to acknowledge the other members of the CancerGrid team and related projects—Daniel Abler, James Brenton, Carlos Caldas, Radu Calinescu, Marta Kwiatkowska, Peter Maccallum, Sylvia Nagl, Aadya Shukla, Matthew Snape, Andrew Tsui, James Welch, Tianyi Zang—for their contributions towards the ideas presented here. We are also very grateful to the Medical Research Council (grant number G0300648), Research Councils UK (grant number EP/F059345/1), the Engineering and Physical Sciences Research Council (grant number EP/H019944/1), the EU Framework Programme 7 (grant number 228436), and Microsoft Research, for funding this work.

Some aspects of the CancerGrid project related to the work described here have also been published elsewhere. An early paper [53] set out the original vision, involving semantic web technology, computational grids, and computer-supported collaborative working; as discussed in this paper, many of these original ideas turned out to be unworkable in practice, and the project soon changed direction. A pair of related papers [54, 55] present formal specifications of an early prototype of our form-based approach, but programmed from scratch in Java rather than by configuring an off-the-shelf application. The CONSORT trials metamodel and the forms-based approach to model-driven engineering are described in a couple of conference papers [24, 18]). The case studies on prospective data collection [23], retrospective data integration [25], and forms metamodelling [37] have each appeared in short papers.

References

[1] CancerGrid website, `http://www.cancergrid.org/`, no date.
[2] The CONSORT Statement, `http://www.consort-statement.org/`, last visited February 2012.
[3] Early Breast Cancer Trialists' Collaborative Group, Tamoxifen for early breast cancer: An overview of the randomised trials, The Lancet 351 (1998) 1451–1467.
[4] University of Birmingham School of Medicine, How Birmingham researchers are taking a measured look at medical treatments, Medlines 4 (1997).
[5] A. Crijns, H. Boezen, J. Schouten, H. Arts, R. Hofstra, P. Willemse, E. de Vries, A. van der Zee, Prognostic factors in ovarian cancer: Current evidence and future prospects, European Journal of Cancer Supplements 1 (2003) 127–145.
[6] P. de Graeff, J. Hall, A. Crijns, G. de Bock, J. Paul, K. Oien, K. ten Hoor, S. de Jong, H. Hollema, J. Bartlett, R. Brown, A. van der Zee, Factors influencing p53 expression in ovarian cancer as a biomarker of clinical outcome in multicentre studies, British Journal of Cancer 95 (2006) 627–633. PubMed identifier 16880779.
[7] J.-F. Lyotard, The Postmodern Condition: A Report on Knowledge, University of Minnesota Press, English edition, 1984.
[8] J. Noble, R. Biddle, Notes on postmodern programming, in: R. Gabriel (Ed.), Proceedings of the Onward Track at OOPSLA 02, pp. 49–71. `http://www.dreamsongs.org/Files/Onward!Proceedings.pdf`.

[9] ISO/IEC JTC1 SC32 WG2, ISO/IEC 11179: Information technology—metadata registries, http://metadata-standards.org/11179/, no date.
[10] CancerGrid metadata registry (cgMDR), http://www.cancergrid.org/index.php?option=com_content&id=8:mdrarticle, no date.
[11] Cancer Data Standards Registry and Repository (caDSR), https://cabig.nci.nih.gov/community/concepts/caDSR/, last visited March 2012.
[12] International Committee of Medical Journal Editors, Uniform requirements for manuscripts submitted to biomedical journals, http://www.icmje.org/urm_main.html, 2010.
[13] Nature Publishing Group, Policy on availability of data and materials, http://www.nature.com/authors/policies/availability.html, 2012.
[14] C. Begg, M. Cho, S. Eastwood, R. Horton, D. Moher, I. Olkin, R. Pitkin, D. Rennie, K. Schulz, D. Simel, D. Stroup, Improving the quality of reporting of randomized controlled trials: The CONSORT statement, Journal of the American Medical Association 276 (1996) 637–639.
[15] D. Moher, K. Schulz, D. G. Altman, The CONSORT statement: Revised recommendations for improving the quality of reports of parallel-group randomised trials, The Lancet 357 (2001) 1191–1194.
[16] K. F. Schulz, D. G. Altman, D. Moher, for the CONSORT Group, CONSORT 2010 statement: Updated guidelines for reporting parallel group randomised trials, PLoS Medicine 7 (2010) e1000251.
[17] D. G. Altman, K. F. Schulz, D. Moher, M. Egger, F. Davidoff, D. Elbourne, P. C. Gtzsche, T. Lang, for the CONSORT Group, The revised CONSORT statement for reporting randomized trials: Explanation and elaboration, Annals of Internal Medicine 134 (2001) 663–694.
[18] C. Crichton, J. Davies, J. Gibbons, S. Harris, A. Tsui, J. Brenton, Metadata-driven software for clinical trials, in: ICSE Workshop on Software Engineering in Healthcare.
[19] Object Management Group, Model driven architecture, http://www.omg.org/mda/, 2001.
[20] C. Poole, H. Earl, NEAT: National breast cancer study of epirubicin plus CMF versus classical CMF adjuvant therapy, http://public.ukcrn.org.uk/search/StudyDetail.aspx?StudyID=643, 2001. ISRCTN 42625759.
[21] C. Poole, H. Howard, J. Dunn, tAnGo: A phase III randomized trial of gemcitabine in paclitaxel-containing, epirubicin-based adjuvant chemotherapy for women with early stage breast cancer, http://public.ukcrn.org.uk/search/StudyDetail.aspx?StudyID=661, 2004. ISRCTN 51146252.
[22] H. Earl, Neo-tAnGo: A neoadjuvant study of sequential epirubicin + cyclophosphamide and paclitaxel ± gemcitabine in the treatment of high risk early breast cancer with molecular profiling, proteomics and candidate gene analysis, http://public.ukcrn.org.uk/search/StudyDetail.aspx?StudyID=1229, 2007. ISRCTN 78234870.
[23] J. Davies, J. Gibbons, S. Harris, J. Metz, A. J. Pollard, M. Snape, Model-driven support for a vaccine study in Kathmandu, in: Microsoft eScience Workshop.
[24] J. Davies, J. Gibbons, R. Calinescu, C. Crichton, S. Harris, A. Tsui, Form follows function: Model-driven engineering for clinical trials, in: International Symposium on Foundations of Health Information Engineering and Systems.
[25] I. Papatheodorou, C. Crichton, L. Morris, P. Maccallum, METABRIC Group, J. Davies, J. D. Brenton, C. Caldas, A metadata approach for clinical data management in translational genomics studies in breast cancer, BMC Medical Genomics 2 (2009) 6.
[26] C. Crichton, Standardization, querying, inference, and validation (SQIV), http://cancergrid.org/index.php?option=com_content&id=25:sqiv, 2009.
[27] Apache Software Foundation, Apache Jena, http://incubator.apache.org/jena/, last visited March 2012.
[28] M. Didonet Del Fabro, P. Valduriez, Towards the efficient development of model transformations using model weaving and matching transformations, Software and Systems Modeling 8 (2009) 305–324.
[29] H. Williams, R. Stern, Prospective clinical trial registration, Journal of Investigative Dermatology 124 (2005) viii–x.
[30] I. Sim, D. E. Detmer, Beyond trial registration: A global trial bank for clinical trial reporting, PLoS Medicine 2 (2005) e365.
[31] C. Laine, R. Horton, C. D. DeAngelis, J. M. Drazen, F. A. Frizelle, F. Godlee, C. Haug, P. C. Hébert, S. Kotzin, A. Marusic, P. Sahni, T. V. Schroeder, H. C. Sox, M. B. V. D. Weyden, F. W. A. Verheugt, Clinical trial registration, British Medical Journal 334 (2007) 1177–1178.
[32] The Cochrane Collaboration, Cochrane Central Register of Controlled Trials, http://onlinelibrary.wiley.com/o/cochrane/cochrane_clcentral_articles_fs.html, last visited March 2012.
[33] J. Golding, M. Pembrey, R. Jones, ALSPAC Study Team, ALSPAC: The Avon Longitudinal Study of Parents and Children. I. Study methodology, Paediatric and perinatal epidemiology 15 (2001) 74–87.
[34] C. Westgarth, J. Liu, J. Heron, A. Ness, P. Bundred, R. Gaskell, A. German, S. McCune, S. Dawson, Dog ownership during pregnancy, maternal activity, and obesity: A cross-sectional study, PLoS ONE 7 (2012) e31315.
[35] C. Winsper, T. Lereya, M. Zanarini, D. Wolke, Involvement in bullying and suicide-related behavior at 11 years: A prospective birth cohort study, Journal of the American Academy of Child and Adolescent Psychiatry 51 (2012) 271–282.e3.
[36] G. Macarthur, M. Smith, R. Melotti, J. Heron, J. Macleod, M. Hickman, R. Kipping, R. Campbell, G. Lewis, Patterns of alcohol use and multiple risk behaviour by gender during early and late adolescence: The ALSPAC cohort, Journal of Public Health 34 (2012) i20–i30.
[37] D. Abler, C. Crichton, J. Welch, J. Davies, S. Harris, Models for forms, in: SPLASH Workshop on Domain-Specific Modeling.
[38] M. Vardigan, P. Heus, W. Thomas, Data Documentation Initiative: Towards a standard for the social sciences, International Journal of Digital Curation 3 (2008) 107–113.
[39] Clinical Data Interchange Standards Consortium, Specification for the Operational Data Model, Technical Report Version 1.3.1, CDISC, 2010.
[40] OpenClinica, http://community.openclinica.com/, last visited March 2012.
[41] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, J. G. Conde, Research electronic data capture (REDCap): A metadata-driven methodology and workflow process for providing translational research informatics support, Journal of Biomedical Informatics 42 (2009) 377–381.
[42] International Cancer Genome Consortium, Esophageal cancer: Esophageal adenocarcinoma, http://www.icgc.org/icgc/cgp/72/508/70708, last visited March 2012.

[43] J. Brenton, J. Davies, J. Gibbons, S. Harris, Accelerating cancer research using semantics-driven technology, in: Microsoft eScience Workshop.
[44] J. Davies, J. Gibbons, S. Harris, D. Warzel, Evolving health informatics: Semantic frameworks and metadata-driven architectures, in: Microsoft eScience Workshop.
[45] T. Blair, Modernising Government, UK Cabinet Office white paper CM 4310, UK Government, 1999.
[46] C. Crichton, J. Davies, J. Gibbons, S. Harris, A. Shukla, Semantic frameworks for e-Government, in: T. Pardo, T. Janowski (Eds.), International Conference on Theory and Practice of Electronic Governance (ICEGOV), pp. 30–39.
[47] J. Davies, S. Harris, C. Crichton, A. Shukla, J. Gibbons, Metadata standards for semantic interoperability in electronic government, in: International Conference on Theory and Practice of Electronic Governance.
[48] C. Crichton, J. Davies, J. Gibbons, S. Harris, A. Shukla, A. Tsui, Semantics-driven development for electronic government applications, in: HICSS Workshop on Electronic Government.
[49] P. Y. H. Wong, J. Gibbons, On specifying and visualising long-running empirical studies, in: International Conference on Model Transformations (ICMT), volume 5063 of *LNCS*, Springer-Verlag, 2008, pp. 76–90.
[50] P. Y. H. Wong, J. Gibbons, Formalisations and applications of BPMN, Science of Computer Programming 76 (2011) 633–650.
[51] P. Y. H. Wong, J. Gibbons, A process-algebraic approach to workflow specification and refinement, in: Software Composition.
[52] P. Y. H. Wong, J. Gibbons, Property specifications for workflow modelling, Science of Computer Programming 76 (2011) 942–967.
[53] J. Brenton, C. Caldas, J. Davies, S. Harris, P. Maccallum, CancerGrid: Developing open standards for clinical cancer informatics, in: UK E-Science All Hands Meeting.
[54] R. Calinescu, Model-based SOA generation for cancer clinical trials, in: L. A. Skar, A. A. Bjerkestrand (Eds.), OOPSLA Workshop on Service-Oriented Architectures, Portland, Oregon, pp. 57–71.
[55] R. Calinescu, S. Harris, J. Gibbons, J. Davies, I. Toujilov, S. Nagl, Model-driven architecture for cancer research, in: Software Engineering and Formal Methods.