WSRF-Based Modeling of Clinical Trial Information for Collaborative Cancer Research

Tianyi Zang[1], Radu Calinescu[1], Steve Harris[1], Andrew Tsui[1]
Marta Kwiatkowska[1], Jeremy Gibbons[1], Jim Davies[1], Peter Maccallum[2], and Carlos Caldas[2]
[1]Computing Laboratory, University of Oxford, Oxford, OX1 3QD, UK
[2]Department of Oncology, University of Cambridge, Cambridge, CB2 0RE, UK

Abstract

The CancerGrid consortium is developing open-standards cancer informatics to address the challenges posed by modern cancer clinical trials. This paper presents the service-oriented software paradigm implemented in CancerGrid to derive clinical trial information management systems for collaborative cancer research across multiple institutions. Our proposal is founded on a combination of a clinical trial (meta)model and WSRF (Web Services Resource Framework), and is currently being evaluated for use in early phase trials. Although primarily targeted at cancer research, our approach is readily applicable to other areas for which a similar information model is available.

1. Introduction

Cancer clinical trials pose significant challenges to the e-Science community [1], [2], as they increasingly take the form of large-scale translational studies, combining the latest laboratory techniques and bioinformatics with traditional statistical analysis based on detailed descriptions of clinical outcomes. The information technology needed to enable this kind of large-scale, collaborative science will need to support syntactic, semantic, and computational interoperability, fast, secure and reliable data transfer, large data sets, including the outputs from new imaging technologies, and complex, multi-disciplinary interaction.

CancerGrid [3], an e-Science consortium project funded by the UK Medical Research Council, and involving five UK universities, is addressing these challenges through the development of metadata-driven, service-oriented technology for cancer informatics. The CancerGrid systems are based on a comprehensive metamodel of cancer clinical trials [4], and use controlled vocabulary services and re-usable, curated metadata elements [5] to enable automatic software generation, interoperability, and data sharing.

In the US, the National Cancer Institute's Center for Bioinformatics (NCICB) has taken a similar approach, although looking at the whole range of cancer informatics requirements. Its cancer Biomedical Informatics Grid (caBIG) initiative is assembling data sets and tools for the creation of a global cancer research infrastructure [6], [7]. The CancerGrid systems interoperate with the existing caBIG architecture, and allow for data and metadata integration across clinical trials operations, demonstrated through a recent collaboration with the Veterans Administration Cooperative Studies Program [8]. However, in current information systems for trial-based cancer clinical research, there is a lack of a standard software paradigm for modeling and managing the clinical trial data.

Cancer research is usually conducted in a Virtual Organization (VO) involving a range of hospitals, clinics, and research institutes. The distributed research VO has a natural mapping to the technologies developed to support Grid computing [9]. Within CancerGrid, model-driven development techniques have been applied to build a model-based, service-oriented Grid architecture for the design, execution and analysis of breast cancer clinical trials [10], [11]. Instances of this architecture are being used in the simulated design and execution of a range of clinical trials [12], [13].

The Web Services Resource Framework (WSRF), driven in part by the Globus Alliance [14], is an open framework for modeling and accessing persistent resources using Web services [15]. WSRF specifications build on existing Web services standards to address the limitation of statelessness inherent to plain Web services through the concept of WS-

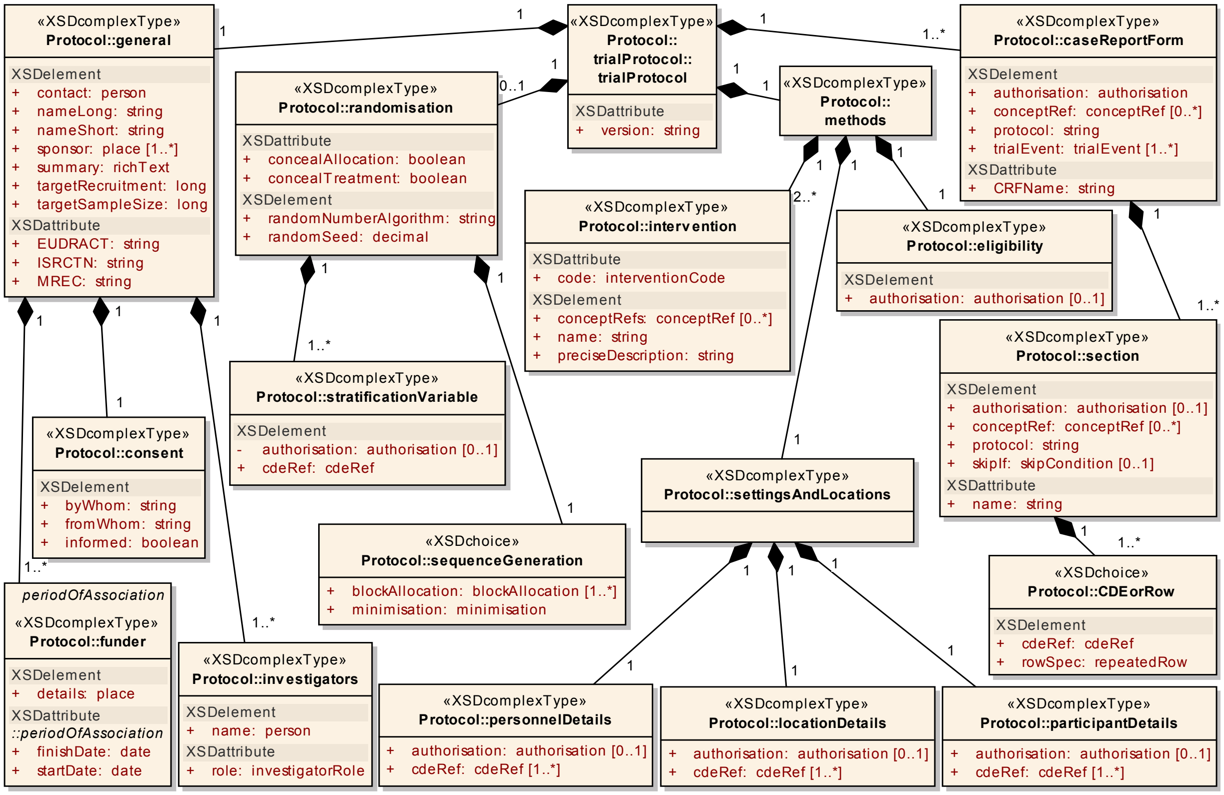

Figure 1. High-level class diagram of the CancerGrid clinical trial metamodel

Resources, and also by defining conventions for the management of state.

A WS-Resource is the combination of a stateful resource and a Web service that interacts with it, and is addressable by an Endpoint Reference (EPR) that adheres to WS-Addressing. WSRF comprises five specifications, explaining how WS-Resources can be represented, accessed, managed, and collected: WS-Resource, WS-ResourceProperties, WS-ResourceLifetime, WS-ServiceGroup, and WS-BaseFaults [15]. WSRF specifications enable the discovery of, introspection upon, and interaction with stateful resources in standard and interoperable ways.

This paper presents a WSRF-based service-oriented software paradigm for the modeling of clinical trials data, implemented as part of CancerGrid. On the basis of the semantics of the clinical trial metamodel, we propose a WSRF-based Service-Oriented Architecture (SOA) and use WSRF in modeling the collection, management and analysis of clinical trial data in collaborative cancer research. The key aspects of our proposal are:

- VO-oriented SOA model and WSRF-compliant Web services are adopted to support the semantics of cancer clinical trials. The use of a formal trial specification allows us to design and develop our clinical trial management system in a concise and unambiguous manner.
- The CancerGrid WS-Resources and WS-Resource groups provide a clinical trial WS-Resource infrastructure, which enables syntactic, semantic, and computational interoperability for collaborative cancer research.
- The WS-Resource based data access control mechanisms and the role-based WS-Resource groups effectively support security-aware data sharing within a cancer research VO.
- The role-based WS-Resource group also makes it easy for a role to organise, manage, and discover a set of role-related WS-Resources.

The Z formal specification language [16] is selected to construct the formal description of the clinical trial metamodel semantics based on our prior experience with using Z in software engineering projects. The availability of a concise and unambiguous specification is essential for a joint project involving multi-disciplinary teams located at several research institutions across the UK. The resulting Z model has been of great help in gaining and conveying a good understanding of trial data model semantics, and in guiding the actual system design and development. The trial data model semantics would have been difficult to specify unambiguously otherwise.

The remainder of the paper is organised as follows. After an overview of our clinical trial metamodel in the next section, Section 3 presents the implied trial data model semantics. Section 4 describes how to model clinical trial data with WSRF. A case study that illustrates the use of the system for two cancer primary clinical trials is provided in Section 5, followed by concluding remarks in Section 6.

2. Clinical trial metamodel

The clinical trial metamodel is built as an embodiment of the Consolidating Standards of Reporting Trials (CONSORT) statement [17], the *de facto* set of guidance rules for the reporting and execution of clinical trials. The CONSORT statement provides a checklist of items that need to be used in writing, reviewing and assessing clinical trial reports, and a flow diagram of the progress through the phases of a clinical trial. The CancerGrid philosophy is that the best way to ensure the availability of all this information for inclusion in clinical trial reports is to specify it explicitly during the design phase of trials.

The CancerGrid clinical trial metamodel [4] reflects this philosophy as shown in Figure 1:

- The general section of a trial design groups together overall data required or recommended by CONSORT, such as the name and a summary of the trial, its registry numbers, sources of funding and contact details.
- The items in the methods section of the CONSORT checklist are captured in the methods and randomisation sections of the model. This includes the interventions (i.e., treatments) for each patient group, the eligibility criteria for patients, the settings and locations for data collection, and a specification of the techniques used in the random allocation of interventions.
- The caseReportForm section of the model defines the Case Report Forms (CRFs) required to support the patient workflow through each stage of a clinical trial. Recruitment, adverse events and other items from the results CONSORT checklist section are being provided for by these CRFs.
- The metamodel enables data sharing between clinical trials by matching all data collected during their execution to well-defined Common Data Elements (CDEs), i.e., controlled sets of cancer concepts and measurements.

3. Implied trial data model semantics

Although UML class diagrams are a powerful design tool, the clinical trial metamodel covers only part of what is required for modeling the clinical trial data. Therefore, the design and development of the clinical trial system also rely on implicit knowledge about cancer clinical trial semantics.

3.1. Common data element

The consistent use of a controlled vocabulary (i.e., a set of domain-specific 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 employed for the generation of small amounts of data [1]. The CancerGrid project is addressing this important requirement by basing its clinical trial model on the use of thesauri (i.e. collections of controlled vocabulary terms and their relationships), and CDEs. A CDE is defined in terms of several basic types [5]:

- CdeID, the set of CDE identifiers used to refer uniquely to specific CDEs.
- CdeType, the set of types that CDE values may have. This supports the syntactic interoperability that allows for data to be moved between different information systems.
- CdeInfo, the metadata that fully define the semantics of the CDE. This supports the semantic interoperability that allows for different data sets to be compared to identify overlapping or related content.

These basic types are summarised below using Z notation [16]:

$$[CdeID, CdeType, CdeInfo]$$

and the CDE type can be specified as:

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

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

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

3.2. Case report forms

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:

$$\begin{array}{l}
\mathit{TrialEvent} ::= \mathit{registration} \mid \mathit{eligibility} \mid \\
\quad \mathit{randomization} \mid \mathit{onStudy} \mid \mathit{treatment} \mid \mathit{offStudy} \mid \\
\quad \mathit{response} \mid \mathit{followUp} \mid \mathit{adverseEvent}
\end{array}$$

Clinicians gather the data corresponding to the trial events by filling in CRFs that comprise CDEs drawn from the cancer CDE repository,

$$\mathit{cancerCdeRep}: \mathit{CdeRepository}$$

A CRF is fully defined by the sequence of trial events corresponding to its sections:

$$\begin{array}{|l}
\multicolumn{1}{l}{\underline{\quad \mathit{CaseReportForm} \quad}} \\
\mathit{events}: \mathrm{seq}\ \mathit{TrialEvent} \\
\hline
\end{array}$$

3.3. Settings and locations

As indicated in [17], the settings and locations where the data were collected affect the external validity of a trial study. All the related information that could influence the observed results should be reported so that readers can assess external validity.

The settings and locations component of the metamodel specifies the data collected about the trial locations (i.e., hospitals or clinical trial units), personnel and patients. These are all combination of CDEs drawn from our metadata repository:

$$\begin{array}{|l}
\multicolumn{1}{l}{\underline{\quad \mathit{SettingsAndLocations} \quad}} \\
\mathit{locationCdeSet}: \mathbb{P}\ \mathit{cancerCdeRep.cdeSet} \\
\mathit{personnelCdeSet}: \mathbb{P}\ \mathit{cancerCdeRep.cdeSet} \\
\mathit{patientCdeSet}: \mathbb{P}\ \mathit{cancerCdeRep.cdeSet} \\
\hline
\end{array}$$

3.4. Trial design

For the purpose of our WSRF-based trial system development, a clinical trial is composed of the *SettingsAndLocations* definition, CRFs of all patients, and the CDEs corresponding to all locations, personnel, patients, and events of each of these patient forms:

$$\begin{array}{|l}
\multicolumn{1}{l}{\underline{\quad \mathit{TrialDesign} \quad}} \\
\mathit{forms}: \mathbb{P}\ \mathit{CaseReportForm} \\
\mathit{eventCdeSet}: \mathit{TrialEvent} \rightarrowtail \mathbb{P}\ \mathit{cancerCdeRep.cdeSet} \\
\mathit{SettingsAndLocations} \\
\hline
\mathrm{dom}\ \mathit{eventCdeSet} = \cup \{ f\!: \mathit{forms} \bullet (\mathrm{ran}\ f.\mathit{events}) \} \\
\forall f_1, f_2: \mathit{forms} \bullet f_1 \neq f_2 \Rightarrow \mathrm{ran} f_1.\mathit{events} \cap \mathrm{ran} f_2.\mathit{even} = \varnothing \\
\hline
\end{array}$$

The location, personnel, and patient instances specific to a clinical trial are defined as:

$$\begin{array}{|l}
\multicolumn{1}{l}{\underline{\quad \mathit{LocationInstance} \quad}} \\
\mathit{id}: \mathbb{N} \\
\mathit{locationDetails}: \mathbb{P}\ (\mathit{Cde} \times \mathit{CdeType}) \\
\hline
\end{array}$$

$$\begin{array}{|l}
\multicolumn{1}{l}{\underline{\quad \mathit{PersonnelInstance} \quad}} \\
\mathit{id}: \mathbb{N} \\
\mathit{locationIds}: \mathbb{P}\ \mathbb{N} \\
\mathit{personnelDetails}: \mathbb{P}\ (\mathit{Cde} \times \mathit{CdeType}) \\
\hline
\end{array}$$

$$\begin{array}{|l}
\multicolumn{1}{l}{\underline{\quad \mathit{PatientInstance} \quad}} \\
\mathit{id}: \mathbb{N} \\
\mathit{locationId}: \mathbb{N} \\
\mathit{patientDetails}: \mathbb{P}\ (\mathit{Cde} \times \mathit{CdeType}) \\
\hline
\end{array}$$

The relationships between the location, personnel, and patient instances in a clinical trial are specified as:

$$\begin{array}{|l}
\mathit{trialLocations}: \mathit{TrialDesign} \longrightarrow \mathbb{P}\ \mathit{LocationInstance} \\
\mathit{trialPersonnel}: \mathit{TrialDesign} \longrightarrow \mathbb{P}\ \mathit{PersonnelInstance} \\
\mathit{trialPatients}: \mathit{TrialDesign} \longrightarrow \mathbb{P}\ \mathit{PatientInstance} \\
\mathit{trialPatientCrfs}: \mathit{TrialDesign} \times \mathit{PatientInstance} \longrightarrow \\
\qquad\qquad \mathbb{P}\ \mathit{CaseReportForm} \\
\hline
\forall t: \mathit{TrialDesign}\ ;\ c: \mathit{Cde}\ ;\ v: \mathit{CdeType}\ ; \\
\ l: \mathit{trialLocations}\ t\ ;\ p: \mathit{trialPersonnel}\ t\ ; \\
\ pt: \mathit{trialPatients}\ t\ ;\ l_1, l_2: \mathit{trialLocations}\ t \bullet \\
\ ((c, v) \in l.\mathit{locationDetails} \Rightarrow c \in t.\mathit{locationCdeSet}) \wedge \\
\ ((c, v) \in p.\mathit{personnelDetails} \Rightarrow c \in t.\mathit{personnelCdeSet}) \\
\ \wedge ((c, v) \in pt.\mathit{patientDetails} \Rightarrow c \in t.\mathit{patientCdeSet}) \\
\ \wedge (l_1.\mathit{id} = l_2.\mathit{id} \Rightarrow l_1 = l_2) \wedge p.\mathit{locationIds} \subseteq \{l.\mathit{id}\} \\
\ \wedge pt.\mathit{locationId} \in \{l.\mathit{id}\} \wedge \mathit{trialPatientCrfs}\ (t, pt) \subseteq t.\mathit{forms}
\end{array}$$

The location is used to group together patients and personnel from the same clinical trial unit or hospital, which is essential when the access to certain data and/or operations is confined to users from the same location. The same patient is not allowed to make multiple registrations at several locations. Personnel are not limited to one location. A patient can be treated by several personnel members, and a personnel member can treat several patients.

The trial data belonging to the patients are gathered using several CRFs. The subset of CDEs that are used to map CRFs to specific patients is specified explicitly in the settings and locations section of the model. The patients for which the forms were completed, the trial personnel that filled them in, and the locations of both classes of people are always stated explicitly in the forms for the trial.

3.5. Authorisation

One of the main objectives in CancerGrid's development of open standards cancer informatics is the enforcement of strict confidentiality constraints associated with cancer clinical trials [1]. This objective is achieved by building the data and operation access policies on top of a role-based access control system [18]. The role-based access control is enforced by the CancerGrid clinical trial model through the inclusion of an *authorisation* element within each part of the model that specifies the sensitive clinical trial data. For instance, the authorisation component associated with a CRF is mandatory, while those of individual CRF parts are optional. The location is used to group together patients and personnel from the same clinical trial unit or hospital so that access to certain data and/or operations is confined to users from the same location.

The set of roles that trial personnel and patients can play is:

$$\begin{array}{l} \mathit{Role} ::= \mathit{patient} \mid \mathit{coordinator} \mid \mathit{clinician} \mid \\ \qquad \mathit{research\_nurse} \mid \mathit{statistician} \end{array}$$

and the set of possible operations available to a specific role is:

$$\mathit{AccessType} ::= \mathit{creation} \mid \mathit{modification} \mid \mathit{querying} \mid \mathit{retrieval}$$

An authorisation-aware trial design defines the rules that specify the CDEs on which each role can perform the allowed operations:

$$\begin{array}{|l} \underline{\quad \mathit{AuthorizedTrialDesign} \qquad\qquad\qquad\qquad} \\ \mathit{trial: TrialDesign} \\ \mathit{accessRules: Role} \times \mathit{AccessType} \mathrel{\rightarrow\!\!\!\!\!+} \mathbb{P}\ \mathit{Cde} \\ \hline \forall\ \mathit{roleCdeSet}\mathit{:}\ \mathrm{ran}\ \mathit{accessRules} \bullet \mathit{roleCdeSet} \\ \quad \subseteq \bigcup \{(\bigcup \{\ \mathit{e: TrialEvent} \bullet (\mathit{trial.eventCdeSet}\ e)\}), \\ \quad (\bigcup\{\mathit{trial.locationCdeSet}\}), (\bigcup\{\mathit{trial.patientCdeSet}\}), \\ \quad (\bigcup\{\mathit{trial.personnelCdeSet}\})\} \\ \hline \end{array}$$

4. Clinical trial resource modeling

4.1. VO-oriented SOA model

Based on the clinical trial semantics described in the previous section, we propose a WSRF-compliant SOA model for a cancer research VO to conduct clinical trial study. As depicted in Figure 2, the layered architecture naturally maps into a cancer research VO.

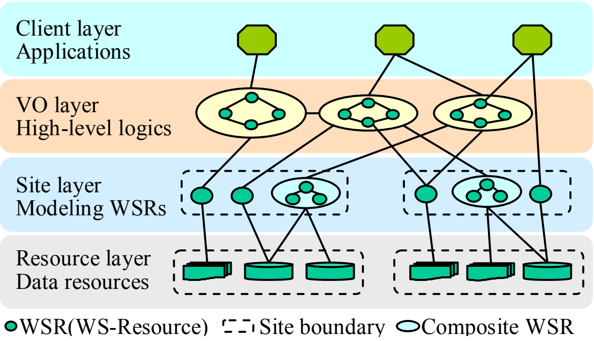

Figure 2. VO-oriented SOA model

From bottom to top there are the CancerGrid data resource layer (Resource layer), the WS-Resource modeling layer (Site layer) and the high-level cancer research logic layer (VO layer). Cancer research features diverse crucial data resources stored and managed at different institutions under different contractual and regulatory environments. The resource layer consists of numerous crucial clinical and tissue based data sources involved in cancer research, such as CancerGrid clinical trial CDE repository, clinical trial and trial data model repositories, databases storing the confidential information pertaining to the clinical trial and the associated staff and patients, etc.

The WS-Resource modeling layer comprises single and composite CancerGrid WS-Resources on each site. The CancerGrid WS-Resources model the states of the underlying data resources and provide a WSRF-compliant standard interface to interact with them. Generally, these modeling WS-Resources are deployed on each site to manage and manipulate the site-specific information and data. The site-specific WS-Resources are behind the firewall and protected within the site domain. Each site and its available WS-Resources need to register with the VO registry. The instantiation and termination of the site WS-Resources can be managed by the services on the VO layer. Especially, in order to support secure access to the underlying resources, the WS-Resource based access control mechanisms are provided on this layer. The entire modeling WS-Resources on this layer form a virtual clinical trial resource level, and serve as the fundamental WS-Resource infrastructure with access control mechanisms for building cancer research business logics.

Based on the modeling WS-Resources infrastructure, the VO layer constructs the high-level business logics for collaborative cancer research by means of Web service composition and choreography technologies. In addition, besides managing and maintaining the memberships according to the VO rules, the VO layer implements the security policies for cancer study by exploiting the WS-Resource based access control mechanisms. The more business logic implemented on the VO layer, the less workload left at the client layer.

4.2. Clinical trial WS-Resources

As specified in the trial data model semantics specification in Section 3, CDEs are the basic units composing the clinical trial data resources: CRFs, settings and locations, and then trial design. The CancerGrid CDEs enable the syntactic and semantic interoperability on the CDE level. Therefore, CDEs are used to define the *resource properties* describing the states of each resource. From the trial model and its semantics, we can produce standard XML schematic definitions for the resource properties by using model-driven technology. Combining them together, we can complete an XML *resource properties document* for the resource. The (meta)model compliant CancerGrid resource properties, together with the resource properties document, support the syntactic and semantic interoperability on the data resource level.

We wrap the resource properties with Web service interfaces defined in the WSRF specifications to implement the clinical trial WS-Resources. The clinical trial WS-Resources enable the computational interoperability as well as the syntactic and semantic interoperability on both the CDE level and the resource level. Six portTypes defined in WS-ResourceProperties [18] can meet the requirements for accessing trial data specified in Subsection 3.5. Each of the portTypes exposes a single operation with the same name as the portType. Using these interfaces we can dynamically retrieve, query, update, insert and delete the resource properties of a WS-Resource at runtime. The details are as below:

- *GetResourcePropertyDocument* allows users to retrieve the values of all resource properties associated with the WS-Resource.
- *GetResourceProperty* allows users to access the value of any resource property by its QName (Qualified Name), i.e., a name including a namespace and a local name.
- *GetMultipleResourceProperties* permits users to access the value of several resource properties at once by their QNames.
- *QueryResourceProperties* permits users to perform complex queries on the resource properties document. XPath can be used as query language.
- *PutResourcePropertyDocument* permits users to completely replace the values of a WS-Resource's properties with a new resource properties document.
- *SetResourceProperties* allows users to request one or several modifications on a service's resource properties. Three actions can be specified by the parameters: *Update*, *Insert* and *Delete*.

4.3. Clinical trial resource sharing

Clinical trial resources are not bound to a single Web service. Multiple Web services can deal with the same WS-Resource instance with different operation logic and from a different role perspective.

In order to support resource sharing, in terms of the access type, the WS-ResourceProperties portTypes can be split into two kinds: read WS-ResourceProperties portTypes and write WS-ResourceProperties portTypes. The former portTypes provide the operations with read permission. The latter provide the operations with write permission. Accordingly, as shown in Figure 3, we provide two kinds of WS-Resources for a resource: read-only WS-Resources for a Web service only implementing read WS-ResourceProperties, and read-write WS-Resources for a Web service implementing both kinds of WS-ResourceProperties.

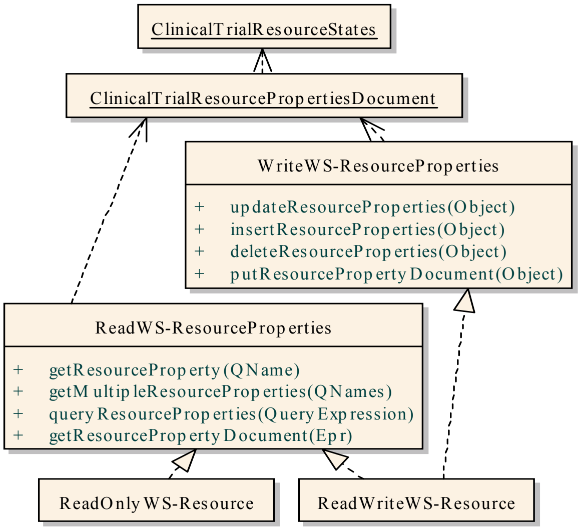
ClinicalTrialResourceStates
ClinicalTrialResourcePropertiesDocument
WriteWS-ResourceProperties
+ updateResourceProperties(Object)
+ insertResourceProperties(Object)
+ deleteResourceProperties(Object)
+ putResourcePropertyDocument(Object)
ReadWS-ResourceProperties
+ getResourceProperty(QName)
+ getMultipleResourceProperties(QNames)
+ queryResourceProperties(QueryExpression)
+ getResourcePropertyDocument(Epr)
ReadOnlyWS-Resource
ReadWriteWS-Resource

Figure 3. Read-only and read-write WS-Resources

Furthermore, we also provide read-only WS-Resources or read-write WS-Resources for parts of the resources properties of a resource for data sharing and access control purposes. A CRF consists of a sequence of event sections. Using the *authorisation* element, each event section in a form may override the form-level access rules with access rules specific to the section. The read-only and read-write WS-Resources can be implemented for each section as well as the entire form.

Similarly, other kinds of WS-Resources for a resource can be implemented, each of which exposes a different set of operations on the resource. The WS-Resource based resource access control mechanisms not only support resource sharing within a single organization, but also allow multiple organizations to work on the same trial resources allowing collaboration within a cancer research VO.

4.4. Clinical trial resource persistence

Obviously there are persistent semantics in WS-Resources for trials-based clinical research. Most modeled resources need to be persisted and maintained in geographically distributed file systems or databases for the quite long period of a cancer study. WSRF does not specify a paradigm to store resource properties persistently. To handle this kind of issue, we implement *load()* and *store()* operations in our WS-Resources. They enable resource properties to be loaded from and persisted to permanent storage. Invoking the two methods on demand, we can make sure the value of a resource property (resource state) in memory is synchronized with the value on disk.

In order to interact with the remote data sources in a Grid environment, OGSA-DAI is adopted as data middleware [19]. OGSA-DAI assists us with access and integration of data across geographically distributed data sources through Web service interfaces for query, update, transform, and delivery of data. Some corresponding OGSA-DAI activities on database or file system are used in the *load()* and *store()* methods.

Persisting the resource to permanent storage always incurs a cost to the application. In order to improve application performance, we apply caching strategies to the practical implementations of the persistent resources. Setting the size and time limit of the cache, we can tradeoff the benefits of persistent resource and their cost. This sort of feature is supported in the Globus implementation of WSRF [20].

4.5. Role-based WS-Resource groups

Cancer research is the collaboration of different roles in highly regulated and complex security environments. For the development of a service-oriented system to support conducting clinical trial study, the following challenges come into focus:

- Each role deals with different kinds of confidential information about the clinical trial. How to effectively limit and control the role's access to the corresponding resources?
- The various resources for a role to process are usually tightly coupled, and due to this interdependency, all of the corresponding WS-Resources must co-exist before the role may interact with them successfully. How to effectively organize and manage the interdependent WS-Resources?
- In order to handle resource states, a role must identify which WS-Resources it should use. How to effectively discover and manage the set of WS-Resources a role is able to use?

On the basis of the WS-ServiceGroup specification, we propose the role-based WS-Resource group to handle the problems mentioned above. The role-based WS-Resource group is implemented as a WS-Resource that groups together EPRs of other WS-Resources meeting the membership rule criteria. The WS-Resource EPRs are generated dynamically, and can be discovered and inspected dynamically. The membership is restricted to only allow the WS-Resources whose access permissions (read-only or read-write) are authorized to the role to join the group. As an example, Figure 4 depicts the WS-Resource groups for coordinator and clinician. To some extent, the role-based WS-Resource group leverages the resource properties access control mechanism described in Subsection 4.3.

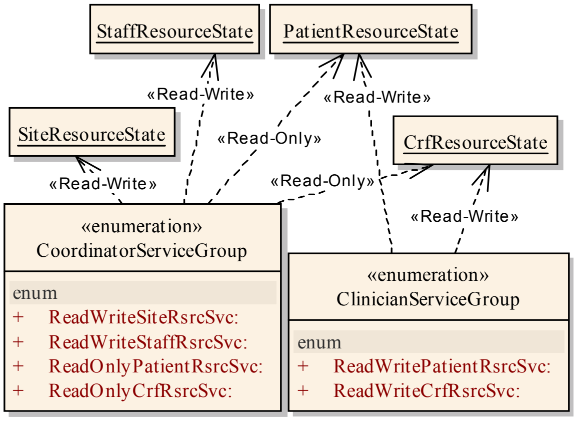

Figure 4. Coordinator and clinician WS-Resource groups

In order to manage and invoke the member WS-Resources in a group, we implement a *GroupFactoryService* on the basis of the factory/instance pattern. A GroupFactoryService serves as a single factory service to expose to the users. The member WS-Resources are deployed behind the firewall to obtain the domain defense as well as enable the integration of internal systems. Upon a request to create an instance of a member WS-Resource, the GroupFactoryService instantiates the member WS-Resources, and then returns the EPR of the created WS-Resource instance to the requestor. The requestor can in turn use the EPR to invoke the operations of WS-Resource to manipulate the resource states. GroupFactoryService is used to manage the logic deciding when and which WS-Resources should be instantiated and destroyed.

As a Web service, the role-based WS-Resource group can also work with WS-Security and other authorisation and authentication technologies to maximize system security. Therefore, the role-based WS-Resource group is not only a self-organized classification mechanism to simplify the discovery and management of sets of WS-Resources, but also a fundamental security mechanism to effectively support role-based access control.

5. Case study

By exploiting the WSRF based service-oriented software paradigm presented in this paper, two prototype implementations of the clinical trial information management systems are developed, respectively, for Neat and tAnGo clinical trials based on Globus Toolkit 4 [20], the *de facto* Grid middleware. They are used to evaluate the effectiveness of our paradigm.

Neat and tAnGo are real cancer clinical trials [12], [13]. They have completed their data acquisition and are currently in the analysis stage. This enables us to conduct simulated executions of the trials based on made-up but realistic patient data, and to demonstrate them to the clinical trial personnel directly involved in running the trials at different cancer study centres. The feedback obtained from trial coordinator, clinicians, and IT staff that took part in the execution of the trials is then used to improve the paradigm of modeling clinical trial data.

The development of WSRF-compliant clinical trial services is based on Neat and tAnGo models, which are instances of the clinical trial metamodel. For each of the two trials, various kinds of WS-Resources with different Web service interfaces are implemented to model the trial data of settings and locations, CRFs, and the trials. Role-based WS-Resource groups are built for the roles specified in the model.

These WS-Resources and WS-Resource groups are then used to implement the various kinds of high-level services for the roles in the two trials, including site management, staff management, patient management, management of various CRFs, trial general information management, and trial data analysis. Each of them provides a different set of operations on the trial data for different roles to get their trial data views.

As a result of the simulated execution of the prototype systems, valuable comments and suggestions for improvement have been obtained from these intended users. The clinical trial modeling WS-Resources, WS-Resource property level access control, and the role-based WS-Resource groups are consistently deemed as the important and effective mechanisms to manipulate the clinical trial data in standard and interoperable ways for building clinical trial business logics. The clinical trial WS-Resources are regarded as the constituent components of the infrastructure in the SOA to enable the syntactic, semantic, and computational interoperability, and effectively support the security-aware data sharing within a cancer study VO. Patient workflow monitoring and management services were reported as insufficient support in the current implementation. The development of these high-level services is a high-priority task in our future work.

6. Conclusion

The VO-oriented SOA model and the approach to model clinical trial data described in this paper represent a WSRF-based service-oriented software paradigm for developing clinical trial management systems supporting collaborative cancer research. This shows how WSRF-compliant Web services can be used to support the semantics of cancer clinical trials. The application of our approach to real clinical trials demonstrates its effectiveness.

The clinical trial WS-Resources utilize the trial CDEs to define clinical trial resource properties and expose different WSRF-compliant Web service interfaces on them. This enables the manipulation of clinical trial data in standard and interoperable ways. The WS-Resource based data access control mechanisms and the role-based WS-Resource groups effectively support security-aware data sharing within a cancer research VO. They can work with other WS-Security technologies to enhance the system security. The role-based WS-Resource group also makes it easy for a role to manage and discover a set of WS-Resources.

The use of a formal trial specification is essential to the design and development of our trial management system in a concise and unambiguous manner. The resulting Z model expresses key properties of the cancer trial data and model on which our system is based. The trial Z specification presented has proved invaluable in improving our understanding of the system and in describing it to other members of the project and intended users.

The developed WS-Resources and WS-Resource groups have provided a clinical trial WS-Resource infrastructure that enables the syntactic, semantic, and computational interoperability to support the implementation of other cancer research logics. Other high-level services, such as a patient workflow management system and cross-trial data analysis services, are being developed within the project. Although the modeled data resources are domain specific, our paradigm can be readily applied to other areas for which a CDE-based information model is available, such as other types of clinical trials. In particular, the security-aware WS-Resource sharing mechanisms are directly applicable to other WSRF-based applications.

Acknowledgment

This work was supported by the UK Medical Research Council grant G0300648. The authors are grateful to Charles Crichton, Peter Brorsson, and the other members of the CancerGrid team for many insightful discussions during the work on the system development.

References

[1] J. Brenton, C. Caldas, J. Davies, S. Harris, and P. Maccallum, "CancerGrid: developing open standards for clinical cancer informatics," in *Proc. UK escience All Hands Meeting 2005*, Nottingham, UK, pp. 678–681.

[2] R. H. J. Begent; J. M. Brady; A. Finkelstein, D. Gavaghan, P. Kerr, H. Parkinson, et al., "Challenges of ultra large scale integration of biomedical computing systems" in *Proc. 18th IEEE Symposium on Computer-based Medical Systems*. Dublin, Ireland, 2005, pp.64−69.

[3] CancerGrid project. Available: http://www.cancergrid.org

[4] S. Harris and R. Calinescu, "CancerGrid clinical trials model 1.1," Oxford University Computing Laboratory, Oxford, UK, CancerGrid Tech. Rep. MRC/1.4.1.3, 2006. Available: http://www.cancergrid.org/public/documents/2006/mrc/Report MRC-1.4.1.3 Clinical trials model 1.1.pdf

[5] I. V. Toujilov and P. Maccallum, "Common data element management architecture," Oxford University Computing Laboratory, Oxford, UK, CancerGrid Tech. Rep. MRC-1.1.2, 2006. Available: http://www.cancergrid.org/public/documents/2006/mrc/ReportMRC-1.1.2CDE management architecture.pdf

[6] US National Cancer Institute. (2007). The cancer Biomedical Informatics Grid. Available: https://cabig.nci.nih.gov

[7] US National Cancer Institute. (2007). The caCORE Software Development Kit. Available: http://ncicb.nci.nih.gov/ infrastructure/cacoresdk

[8] Veterans Administration Cooperative Studies Program. Available: http://www.vacsp.gov

[9] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations", *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 200-222, 2001.

[10] R. Calinescu, S. Harris, J. Gibbons, J. Davies, I. Toujilov, and S. Nagl, "Model-driven architecture for cancer research," in *Proc. 5th IEEE Int. Conf. on Software Engineering and Formal Methods*, London, 2007, pp. 59-68.

[11] R. Calinescu, T. Zang, and P. Maccallum, "CancerGrid Architecture Development," Oxford University Computing Laboratory, Oxford, UK, CancerGrid Tech. Rep. MRC-3.2.3, 2006. Available: http://www.cancergrid.org/public/documents/2006/mrc/ReportMRC-3.2.3.pdf

[12] C. Poole and H. Earl. NEAT: National breast cancer study of epirubicin plus CMF versus classical CMF adjuvant therapy. Available: http://www.ncrn.org.uk/portfolio/dbase.asp

[13] C. Poole, H. Howard, and J. Dunn. (2003). tAnGo: A phase III randomized trial of gemcitabine in paclitaxel-containing, epirubicin based adjuvant chemotherapy for women with early stage breast cancer. Available: http://www.isdscotland.org/isd/files/tAnGo protocol version 2.0 July 2003.pdf

[14] Globus Alliance. Web Service Resource Framework. Available: http://www.globus.org/wsrf

[15] OASIS WSRF v1.2 standard. Available: http://www.Oasis-open.org/committees/tc_home.php?wg_abbrev=wsrfWeb

[16] J. Woodcock and J. Davies, *Using Z. Specification, Refinement and Proof.* Prentice Hall, 1996.

[17] D. G. Altman, K. F. Schulz, D. Moher, M. Egger, F. Davidoff, D. Elbourne, P. C. Gotzsche, and T. Lang, "The revised CONSORT statement for reporting randomized trials: explanation and elaboration," *Annals of Internal Medicine*, vol. 134, no. 8, pp. 663–694, 2001.

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

[19] OGSA-DAI. Available: http://www.ogsadai.org.uk

[20] Globus Toolkit 4. Available: http://www.globus.org/toolkit