Software Architecture Document

Clinical Research Chart Cell 1.3
Abstract:

This is a software architecture document for CRC (Clinical Research Chart) cell. It identifies and explains the important architectural elements. This document will serve the needs of stakeholders to understand the system concepts, and give a brief summary of the use of the CRC message format.

Revision History

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Date</th>
<th>Made By</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>7.13.07</td>
<td>Rajesh Kuttan</td>
<td>Initial content</td>
</tr>
<tr>
<td>1.1</td>
<td>7.02.08</td>
<td>Rajesh Kuttan</td>
<td>SQL Server 2005 support.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Access to multiple data mart based on the domain-id, project-id and user-id.</td>
</tr>
</tbody>
</table>
# Table of Contents

Revision History .............................................................. 2

1. Overview ........................................................................ 4
   1.1. CRC Definitions, Acronyms and Abbreviations ...................... 4
   1.2. User Role .................................................................. 5
   1.3. Security ..................................................................... 5
   1.4. Scope of the system ..................................................... 5
   1.5. Assumptions/Constraints ............................................... 5
   1.6. Technical Platform: .................................................... 5
      1.6.1. Transaction .......................................................... 6
      1.6.2. Security ............................................................... 6
      1.6.3. Persistence ........................................................... 6
      1.6.4. Reliability/Availability ............................................. 6
      1.6.5. Performance .......................................................... 6
   2. Use Case ....................................................................... 7
      2.1. Use Case — Run Query from Definition .............................. 7
      2.1.1. CRC Query execution using Queue Model ....................... 8
      2.1.1.2. Context Diagram ................................................. 9
      2.2. Use Case — Get PDO/Table PDO from PatientSet ................. 9
   3. Architecture Description .................................................. 9
      3.1. Components and Connector View .................................... 10
      3.1.1. Client-Server View .................................................. 10
      3.1.1.1. Primary Presentation ............................................. 10
      3.1.1.2. Element Catalog .................................................. 11
      3.1.1.3. Design Rationale, Constraints ................................. 13
      3.2. Module View type ..................................................... 13
      3.2.1. Decomposition Style ............................................... 13
         3.2.1.1. Primary Presentation ........................................... 13
         3.2.1.2. Element Catalog ............................................... 14
         3.2.1.3. Relations and their properties .............................. 14
         3.2.1.4. Context Diagram ................................................. 14
      3.2.2. Uses Style .......................................................... 14
         3.2.2.1. Primary Presentation ........................................... 14
         3.2.2.2. Element Catalog ............................................... 15
         3.2.2.4. Context Diagram ................................................. 16
      3.3. Mappings of Styles ................................................... 16
   6. Data View .................................................................... 17
      6.1. Volumes ................................................................. 17
   7. Deployment View ......................................................... 18
      7.1. Global Overview ...................................................... 18
      7.2. Detailed deployment model ......................................... 18
   References .................................................................... 19
1. Overview

The Clinical Research Chart (CRC) repository cell is one of the core cells in the i2b2 Hive. The CRC cell is designed with several requirements. The main requirements are:

1. It must be able to hold healthcare information from many different venues and allow it to be queried rapidly even if there are hundreds of millions of rows.
2. It must be easily combined with other project repositories to form large unified repositories.
3. Finally, it must allow objects to be stored that are present in the genomic data.

Currently information in the CRC cell is related to clinical data and hence it’s also called Clinical Research Chart. For the remainder of this document, the terms CRC and data repository cell will be used interchangeably to refer to the same cell. CRC is a data warehouse of patient’s phenotype and genotype information. It is supported by a powerful metadata management module (the Ontology Cell). Currently the Clinical Research Chart handles concepts such as diagnoses, procedures, medications, and lab tests, but the structure of the table gives enough flexibility to expand this to include virtually any kind of observation. The presence of both genotype and phenotype information makes this cell a powerful tool for researchers. All patient data present in the CRC are de-identified, except the real patient notes from hospitals. These notes are stored in encrypted form, so they may be only viewed by users enabled with an encryption key.

1.1. CRC Definitions, Acronyms and Abbreviations

1.1.1 Patient Data Object (PDO):
This Object mirrors the star schema database model of the data mart. It holds patient information such as clinical observations, demographics and provider data.

1.1.2 Setfinder Query:
Setfinder queries are used to create a set of patients that satisfy a criteria presented in the query. The setfinder query is composed of query constraints, like the list of Panels and its Items.
1.1.3 Observation Fact:
Any observation made on a Patient can be stored as fact information in CRC data mart. The user can fetch this fact information via the PDO queries.

1.2 User Role
The primary roles/participants in the CRC system are as follows:
- User – Create queries and access them only if he/she is owner to of the query.
- Project Investigator – Create queries and can access queries created by different users with in the project.

1.3 Security
Users can accesses the CRC with domain-id, project-id, user-id and password combination, which is authenticated through the Project Management Cell. The implementation detail of Project Management Cell is considered out-of scope to this system context.

1.4 Scope of the system
Some other participants, currently outside the scope of CRC, are:
- Project Management Cell
- Ontology Cell
- edu.harvard.i2b2.common

1.5 Assumptions/Constraints
- The data in the CRC Datamart DB will not have identified data. Exception to this is the Patient Notes stored inside “OBSERVATION_BLOB” which will be encrypted.
- Client will make “Patient Data Object Query/Request” in multiple requests if the input list(PatientSet or ObservationSet) is big.

1.6 Technical Platform:
The technology used to build the product is as follows
- Java 2 Standard Edition 5.0
- Oracle Server 10g database
- SQLServer 2005
- Xerces2 XML parser
- JBoss Application server version 4.2.2 and higher
• Spring Web Framework 2.0
• Axis2.1 web service (SOAP/REST)

1.6.1 Transaction
The CRC system is transactional, leveraging the technical platform capabilities. The transaction management model of the J2EE platform will be reused intensively. Note: In current implementation, to support long running setfinder queries, transaction management will be manually turned off until the completion of the query.

1.6.2 Security
The application must implement basic security behaviors:
• Authentication: Authenticate using the combination of domain id, project id, user name and a password.
• Authorization: Based on the user role, the user may access setfinder queries created by other users, view patient notes, etc.
• Confidentiality: Sensitive data must be encrypted (Patient Notes)
• Data integrity: Data sent across the network cannot be modified by a tier
• Auditing: In the later releases we might implement logging of very sensitive actions

1.6.3 Persistence
Application uses the JDBC calls to persist data.

1.6.4 Reliability/Availability
The Reliability/Availability will be addressed through the J2EE platform
Targeted availability is 16/7: 16 hours a day, 7 days a week
The time left (8 hours) is reserved for any maintenance activities

1.6.5 Performance
The user authentication with project management cell must be under 10 seconds.
The concept code lookup to the ontology cell must be under 10 seconds.
2. Use Case

The diagram below depicts the common use cases a user can perform with the CRC cell.

2.1 Use Case — Run a Query from Panel Definition

- Validate the user by calling the Project Management Cell.
- Select a data mart based on the combination of domain-id, project-id and user-id.
- Call Ontology Cell with the item key and determine the dimension table to join with the fact table.
- Save the query panel definition and the generated Sql statements.
- Generate the list of output like the patient count, patient gender count, patient set, etc.
- To scale the application and to support long running Sql, the execution of Sql is handled inside a set of queues. At first the query Sql statements will be executed inside a small job queue, if it didn’t complete with in the certain time period, then the jobs will be transferred to mid size job queue and then to large size job queue.
If the Sql execution completes before the “result_waittime_ms” which is specified in the request, then the query results is passed in the response message, otherwise the status of the query is passed in the response message.

2.1.1 CRC Query execution using Queue Model:

2.1.1.1 Sequence Diagram
2.2 Use Case — Get PDO from PatientSet

- Validate the user via the Project Management Cell
- Select the data mart based on the domain-id, project-id and user-id.
- Call Ontology Cell with the item key and determine the dimension table to join with the fact table.
- Using the given patient set or Observation set, apply the Panel filters and return PDO.

3. Architecture Description

As noted in “Documenting Software Architectures”, software architecture is a complex entity that cannot be described in a simple one-dimensional fashion. This document provides the description of the architecture as multiple views. Each view conveys the different attributes of the architecture.

1) Components and Connector View
   a) Client-Server Style

2) Module View
   a) Decomposition Style
   b) Uses Style

3) Data View

4) Deployment View
3.1 Components and Connector View

A Component and Connector view represents the runtime instances and the protocols of connection between the instances. The connectors represent the properties such as concurrency, protocols and information flows. Following diagram represents the Component and Connector view for the multi-user installation. As seen below, component instances are shown in more detail with specific connectors drawn in different notations.

3.1.1. Client-Server View

The CRC system is represented using the C&C Client-Server view.

3.1.1.1 Primary Presentation

![Diagram of Component and Connector View]
3.1.1.2 Element Catalog

3.1.1.2.1 Elements and their Properties

Properties of CRC cell elements are:
- Element Name, given in the following table
- Whether the element is a data repository, a data accessor, a communication method, a query, a client component or a server component
- A description of the element

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webservice Client</td>
<td>Client</td>
<td>Webservice client (i2b2 Workbench/Navigator) submits the requests to CRC Server components and renders response XML.</td>
</tr>
<tr>
<td>CRC Server</td>
<td>Server</td>
<td>Provides Web Service Interface for the CRC system. It supports both SOAP and REST protocols. It uses Project Management server to handle user authentication. It uses Ontology server to lookup the concepts metadata. Select the CRC data mart based on domain-id, project-id and user-id. It stores Setfinder query definition, query run instance and the corresponding query results. The user can then request Patient Data Object using the Setfinder results.</td>
</tr>
<tr>
<td>Project Management Server</td>
<td>Server</td>
<td>CRC cell uses Project Management cell to</td>
</tr>
</tbody>
</table>
authenticate user. CRC cell constructs PM Cell request message and makes a web service call to Project Management Cell.

| Ontology Server | Server                  | CRC sends web service requests to the Ontology cell to get metadata information about an Observation fact’s concepts. |
|-----------------|-------------------------|----------------------------------------------------------------------------------------------------------------|---|
| CRC Datamart DB | Data repository         | This repository is mainly a data mart for patient’s clinical observation information represented in star schema. The Server supports multiple data marts. And the data marts are selected based on the domain-id, project-id and user-id combination. This database also holds CRC user’s queries (Setfinder query) information and its results like patient sets, visit sets, etc. |
| Full SQL        | Query Connector         | SQL query used as a connector between the CRC System and the CRC Datamart DB. |
| Web Service     | Request Connector       | SOAP or REST request used to communicate with the external system. |

### 3.1.1.2 Relations and Their Properties

The relation of this C&C view is *attachment*, dictating how components and connectors are attached to each other. The relations are as shown in the primary presentation; there are no additional ones.
3.1.1.3 Design Rationale, Constraints

**N-tier Architecture**

The client-server style depicts the n-tier architecture that separates presentation layer from business logic and data access layer thus providing for a high degree of portability through the application of the principle of Separation of Concerns.

3.2 Module View type

The module view shows how the system is decomposed into implementation units and how the functionality is allocated to these units. The layers show how modules are encapsulated and structured. The layers represent the “allowed-to-use” relation.

The following sections describe the module view using Decomposition and Uses Style.

3.2.1 Decomposition Style

The Decomposition view presents the functionality in terms of manageable work pieces. They can be further decomposed to present higher level of details. The decomposition view identifies modules and breaks them down into sub-modules and so on, till a desired level of granularity is achieved. The “Uses” style shows the relationships between modules and sub-modules. This view is very helpful for implementation, integration and testing the system.

3.2.1.1 Primary Presentation

<table>
<thead>
<tr>
<th>System</th>
<th>Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC</td>
<td>Setfinder Manager</td>
</tr>
<tr>
<td></td>
<td>PDO Manager</td>
</tr>
</tbody>
</table>
3.2.1.2 Element Catalog

Elements and their properties

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setfinder Manager</td>
<td>Subsystem</td>
<td>This subsystem manages user’s Setfinder queries. Keep tracks of query information like query definition, its Sql, owner of query, etc. Also the results of query like the patient set, visit set, etc is stored.</td>
</tr>
<tr>
<td>PDO Manager</td>
<td>Subsystem</td>
<td>This manages both plain and table Patient Data object queries.</td>
</tr>
</tbody>
</table>

3.2.1.3 Relations and their properties

The subsystem elements form the *is-part* of relation with the overall CRC system.

3.2.1.4 Context Diagram

![Context Diagram](image)

3.2.2 Uses Style

3.2.2.1 Primary Presentation

<table>
<thead>
<tr>
<th>System</th>
<th>Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC</td>
<td>CRC Module</td>
</tr>
<tr>
<td>Setfinder Manager Subsystem</td>
<td>Setfinder Web Service</td>
</tr>
<tr>
<td></td>
<td>Setfinder EJB</td>
</tr>
<tr>
<td></td>
<td>Setfinder DAO</td>
</tr>
<tr>
<td></td>
<td>edu.harvard.i2b2.common</td>
</tr>
<tr>
<td>PDO Manager Subsystem</td>
<td>PDO Web Service</td>
</tr>
</tbody>
</table>
### 3.2.2.2 Element Catalog

**Elements and their properties**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC Module</td>
<td>Module</td>
<td>User Login Module authenticates through PIN Server System with user id and PIN.</td>
</tr>
<tr>
<td>Setfinder Webservice</td>
<td>Module</td>
<td>Provides web service interface to Setfinder operations.</td>
</tr>
<tr>
<td>Setfinder EJB</td>
<td>Module</td>
<td>Delegates Setfinder requests to DAO layer to perform database operations.</td>
</tr>
<tr>
<td>Setfinder DAO</td>
<td>Module</td>
<td>Supports operation like create query master, delete query, saving query definition and its results.</td>
</tr>
<tr>
<td>PDO Webservice</td>
<td>Module</td>
<td>Provides web service interface for PDO requests.</td>
</tr>
<tr>
<td>PDO EJB</td>
<td>Module</td>
<td>Module to delegate PDO requests to corresponding PDO and to build PDO response message.</td>
</tr>
<tr>
<td>PDO DAO</td>
<td>Module</td>
<td>Module to query database based on PDO requests.</td>
</tr>
<tr>
<td>edu.harvard.i2b2.common</td>
<td>Module</td>
<td>This module provides utility classes to handle JAXB, JNDI, etc.</td>
</tr>
<tr>
<td>Persistence Service</td>
<td>Module</td>
<td>Provides SQL interface to database.</td>
</tr>
</tbody>
</table>

### 3.2.2.3 Relations and their properties

The modules in this style follow a *depends-on* relation.
3.2.2.4 Context Diagram

3.3 Mappings of Styles

The following table is a mapping between the elements in the Component & Connector Client-Server view shown in section 4, and the Modules Uses view and Decomposition view shown in sections 5 and 6.

The relationship shown is *is-implemented-by*, i.e. the elements from the C&C view shown at the top of the table are implemented by any selected elements from the Modules views, denoted by an “X” in the corresponding cell.
6. Data View

The key data elements related to the CRC system are:

6.1. Volumes:

- Estimated new Setfinder query: 100 a day, with peaks in the morning
- Average PatientSet size 100,000
- CRC registered individual user: about 150
7. Deployment View

7.1 Global Overview

7.2 Detailed deployment model
References


i2b2 (Informatics for Integrating Biology and the Bedside) https://www.i2b2.org/resrcs/hive.html