

# **OSCAR Clinical Coding Project**

## **Final Report**

**Provided by Leverage Analytics**

**Box 49017, Victoria BC, V8P 5V8**

**Fieran Mason-Blakley**

**[fieranmason@gmail.com](mailto:fieranmason@gmail.com)**

**To Robyn Kuropatwa Ltd.**

# Contents

---

<b>1</b>	<b>Executive Summary</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
<b>3</b>	<b>The RKL GitHub Page</b>	<b>6</b>
3.1	OSCAR Deployment . . . . .	7
3.1.1	Installation for Demonstration (By Package) . . . . .	7
3.1.2	Installation for Quick and Dirty Development (By Script) . . . . .	9
3.1.3	Installation for Substantive Development (Manual) . . . . .	9
3.2	The CPT Service . . . . .	9
3.3	The CPT Service API . . . . .	11
<b>4</b>	<b>OSCAR2 at Canadian Webhosting</b>	<b>11</b>
4.1	An OSAR 15 BC Beta Deployment . . . . .	11
4.2	The Clinical Coding a Prototype . . . . .	13
4.3	The Deployment of Dennis Lee's EForm . . . . .	17
<b>5</b>	<b>Development Challenges and Solutions</b>	<b>17</b>
5.1	Maintenance Tips . . . . .	17
5.1.1	Locked out? . . . . .	17
5.1.2	Want to Stop Account Expiry? . . . . .	18
5.2	Development Tips . . . . .	18
5.2.1	Gerrit Patch Instructions . . . . .	18
5.2.2	Building OSCAR with and without unit tests . . . . .	19
5.2.3	solving SQL Grammar Exception . . . . .	19
5.2.4	Accepting Self-Signed Certs . . . . .	19
5.2.5	Working with Surefire . . . . .	19
5.2.6	Spring Applications on Ubuntu 12.04 LTS are a Problem . . . . .	20
5.2.7	What version of Ubuntu am I running? . . . . .	20
5.2.8	What version of Java is Tomcat running . . . . .	20
5.2.9	Leveraging a MySQL connector in a WAR . . . . .	20
<b>6</b>	<b>Recommendations and Conclusions</b>	<b>21</b>
6.1	Recommendations . . . . .	21

6.2 Conclusions . . . . .	22
---------------------------	----

# 1 EXECUTIVE SUMMARY

---

In the fall of 2016 RKL Ltd. engaged Leverage Analytics in the OSCAR Clinical Coding project to explore the effort required to enhance clinical coding functionality in commercial EMRs. This was a subproject of RKL's Nova Scotia Physician's Manual Modernization Project (PMMP) engagement with the Nova Scotia Ministry of Health and Wellness (MHW). The project used the OSCAR open source EMR as the basis for this exploration. The project fell short of the desired outcomes.

The reasons for these shortcomings are strongly associated with the technical debt in the OSCAR EMR. Unfortunately, the jurisdictional elements of OSCAR are not well modularized. These same jurisdictional elements are closely tied to the clinical coding terminologies used. As a consequence, only limited progress was made in the development of clinical coding features.

What was accomplished was the development of a CPT clinical coding service; a clinical coding prototype which uses the CPT Service and Dennis Lee's SNOMED CT terminology service; and an OSCAR deployment which uses the latest code as of Nov 22 2016 for OSCAR 15 BETA. The CPT Service, the prototype and the OSCAR 15 build have been deployed on RKLs [OSCAR2](#) webserver at Canadian Webhosting. They are also provided on the RKL [GitHub site](#),

The key recommendations which arise from this work are the following:

1. If fully integrated feature development in OSCAR is to be undertaken, a budget sufficient to improve jurisdictional modularization is required.
2. If superficial integration with OSCAR is sufficient then interfaces like EForms or the Web-services package (ws) should be explored. Each of these approaches will result in what are essentially "connected" applications.
3. If "connected applications" are to be funded by RKL, an experienced Java web developer is necessary. They must have comprehensive knowledge of a significant breadth of the technologies used in the OSCAR technical stack including Spring, Struts and Hibernate. They must also be well versed in the Gerrit/Jenkins continuous integration (CI) solution. Finally, knowledge of the HL7 CDA standard is likely to become an important skill set early in the development process. Significant Linux system administration skill will be a valuable asset in debugging the wide range of configuration issues which will arise during development.

## 2 INTRODUCTION

---

In the fall of 2016 RKL Ltd. engaged Leverage Analytics in the OSCAR Clinical Coding project to explore the effort required to enhance clinical coding functionality in commercial EMRs. This was a subproject of RKL's Nova Scotia Physician's Manual Modernization Project (PMMP) engagement with the Nova Scotia Ministry of Health and Wellness (MHW). In the broader context of this parent project, RKL had been charged with identifying requirements for clinical coding in EMRs for the province of Nova Scotia. As part of this work, RKL was to provide guidance on the level of expected effort to implement the features specified in the provided requirements.

Leverage Analytics was asked to implement prototypes which demonstrated the integration of clinical coding functionality within the OSCAR EMR. OSCAR was chosen as it was an open source commercial project and thus access to the source code was available for the prototyping of these new features. It was hoped that both data entry and extraction functionality could be developed, but this proved more challenging than the value of the effort warranted. The reasons for this shortcoming are discussed more fully in the recommendations section of this report (Section 6).

What was accomplished was the development and deployment of a CPT REST service, the development of an accompanying Java based API module, and the integration of that Java based API module as well as an existing SNOMED terminology API (developed by Dennis Lee) into OSCAR's BC Billing module. The BC Billing module was chosen as a starting point for exploration because of Leverage's familiarity with the BC system. This allowed for the demonstration of early results thus quickly addressing project risks by investigating the crosscutting challenges of full feature integration early. The BC Billing Module eventually proved problematic as the billing modules in OSCAR are not well modularized and therefore, a distinct Nova Scotia sandbox module could not be developed to isolate the prototype features from the other billing modules in OSCAR.

Through the remainder of this document, we will discuss

1. The RKL [Github site](#) - the primary platform where the documentation and technical deliverables have been provided to RKL (Section 3) and the deliverables provided there:
  - (a) OSCAR Installation Instructions (Subsection 3.1)
    - i. For clinical use/ demonstration (Subsubsection 3.1.1)
    - ii. For quick and dirty development (Subsubsection 3.1.2)
    - iii. For substantive development (Subsubsection 3.1.3)

- (b) The CPTService (Subsection 3.2)
  - (c) The CPTService API (Subsection 3.3)
2. The Canadian Webhosting OSCAR2 server - the secondary platform where deliverables have been provided to RKL (Section 4) and the deliverables provided there:
    - (a) A deployment of the current OSCAR 15 BC Beta (Subsection 4.1)
    - (b) The Clinical Coding prototype (Subsection 4.2)
    - (c) Inclusion of Dennis Lee's EForms in these two deployments (Subsection 4.3)
  3. Development and deployment issues and their solutions (Section 5)
  4. Conclusions and recommendations for future work (Section 6)

### 3 THE RKL GITHUB PAGE

---

The source documentation for the RKL OSCAR Clinical Coding project was recorded on the [RKL Github site](#). Github was chosen as it is a popular source control site for software developers which provides free online Git repositories. Git is the source control system used by OSCAR EMR. The choice of Github as a delivery platform allowed for documentation and technical deliverables to be provided on a singular central platform designed for this purpose and which will be familiar to subsequent developers should RKL decide to pursue the investigation tackled in the project again.

Dennis Wellborn with Github username dlwellborn was made a manager of this resource during the transition of the project from Leverage Analytics to RKL. Dennis Wellborn was chosen as the technical successor as he is the most technical of the ongoing contractors engaged with RKL and most likely to manage any continuing aspects of this project moving forward. The site provides detailed reporting of the information that this report summarizes. The site is publicly [available](#). In order to edit the site however, it will be necessary to create a Github account. In order to manage the site, it will be necessary to have Dennis Wellborn add a Github user as an Owner of this site.

The deliverables provided here are:

1. This report.
2. An introduction to OSCAR deployment (Subsection 3.1)

3. Instructions on how to deploy OSCAR using the Debian package provided by OSCAR EMR. These instructions are for deploying a demo OSCAR server - they are likely good enough for production use, but are not sanctioned as such by the OSCAR EMR Quality Management System (Subsubsection 3.1.1).
4. A script for deploying OSCAR initially developed by the LEAD lab at the University of Victoria and modified by Leverage Analytics. This script will get OSCAR up and running and will also provide a user with code level access to OSCAR, but with minimal support for development (Subsubsection 3.1.2).
5. Manual installation instructions for OSCAR which are recommended for anyone intending on pursuing serious development of the OSCAR EMR (Subsubsection 3.1.3).
6. HelloWorldService - a basic Spring REST service which provides developers new to Spring a jumping off point for developing new REST services which might provide access to alternative terminologies.
7. The CPTService - a webservice developed to support the use of CPT codes in place of MSP service codes in the OSCAR EMR.
8. The CPTServiceApi - an interface project intended to be incorporated into Java based projects that leverage the CPTApi.

## 3.1 OSCAR DEPLOYMENT

The OSCAR EMR is a Java based web application which uses a MySQL database. It is recommended that OSCAR be deployed on a Tomcat webserver using a Ubuntu 14.04 LTS linux platform.

### 3.1.1 INSTALLATION FOR DEMONSTRATION (BY PACKAGE)

The OSCAR Debian install instructions are provided by OSCAR EMR in the OSCAR manual. The instructions advise that a conservative Linux distribution like Debian Jessie be used for production use. Leverage Analytics tested the provided instructions instead on Ubuntu 14.04.03 amd64 which is also relatively conservative. The instructions provided are for local deployment using Virtual Box on a Mac OSX platform. The installation steps are provided below:

1. Create a new Virtual Box Linux virtual machine (VM).

2. Mount a new optical drive for this machine with a Ubuntu 14.04 iso. ubuntu-14.04.3-desktop-amd64.iso was used in our trial installation.
3. We did not enable encryption options as we were not working with a production deployment
4. We did enable updates during install
5. We chose the LVM option
6. The Ubuntu installation took ~5 minutes
7. **DON'T UPGRADE**. This will upgrade Ubuntu to 16.04LTS and will cause problems with the OSCAR installation
8. Install Guest Additions for your guest Ubuntu VM - host toolbar->Devices->Insert Guest Additions CD Image
9. Run Additions
10. Reboot the VM
11. Adjust screen resolution
12. Enable share clipboard - Devices -> Shared Clipboard -> Bidirectional
13. Enable drag and drop - Devices -> Drag and Drop -> Bidirectional
14. Click the Ubuntu Icon at the top of the guest taskbar
15. Search for terminal
16. Begin OSCAR **installation**
  - (a) install java as per instructions
  - (b) install vim so you can edit your /etc/profile
  - (c) install wkhtmltopdf as per instructions
  - (d) install tomcat and mysql as per instructions
  - (e) install deb as per instructions
    - The package installer figures out your region dynamically
    - In the trial installation demo data was installed



- Access OSCAR from a browser in the Guest system - <https://localhost:8443/oscar/index.jsp>  
– login - oscar/doc/mac2002/1117

17. The whole process should be complete in ~1-1.5hours

### 3.1.2 INSTALLATION FOR QUICK AND DIRTY DEVELOPMENT (BY SCRIPT)

During the **SCOOP Health project**, Raymond Rusk, the OSCAR EMR Quality Management System (QMS) office as of the writing of this document, developed automated scripts to deploying OSCAR and building it from source. Leverage Analytics tested deployment using this script and found that the original script was out of date and required modification. These modifications were made and the updated script is now available for RKL use.

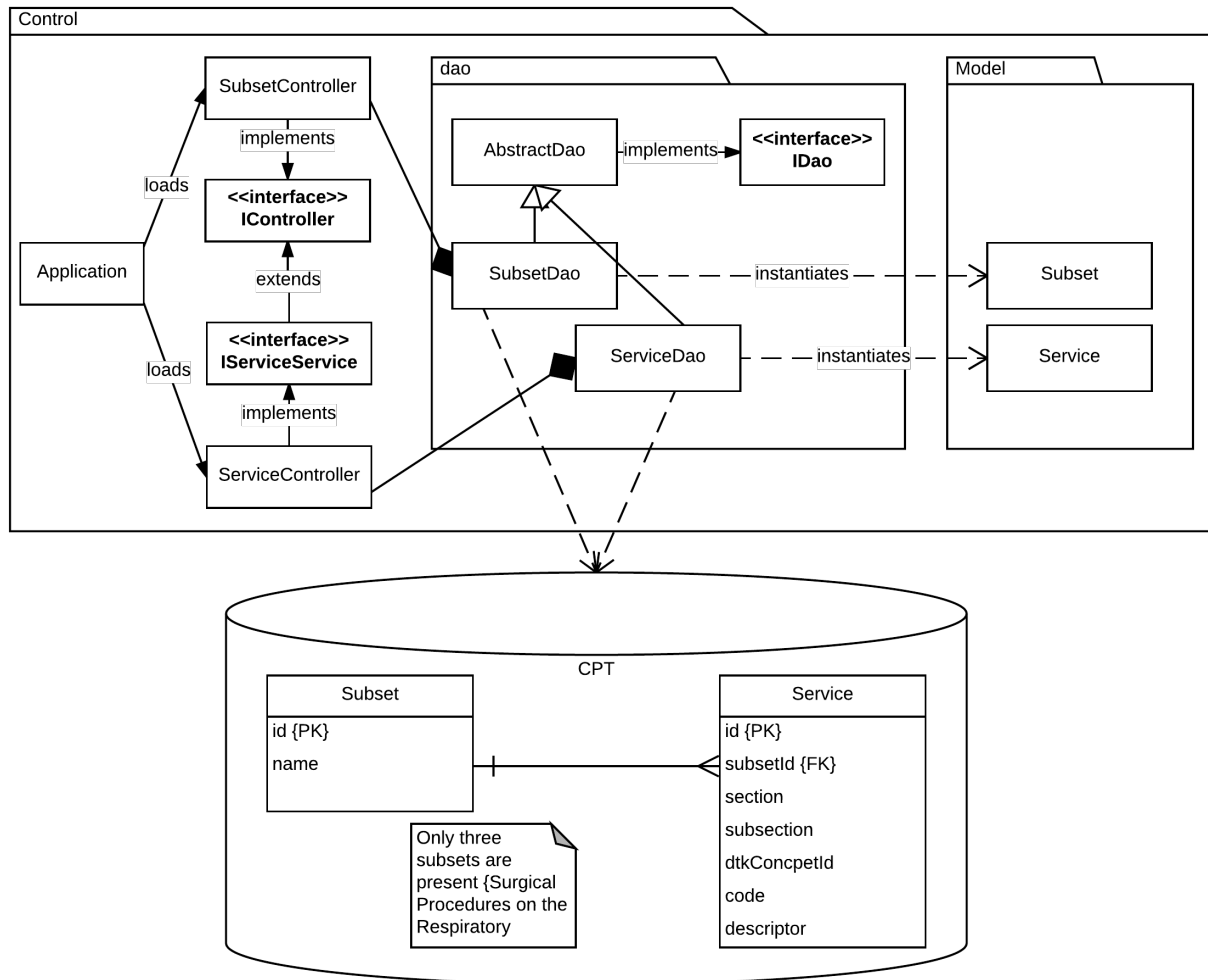
### 3.1.3 INSTALLATION FOR SUBSTANTIVE DEVELOPMENT (MANUAL)

The **SCOOP Health project** developed a strong **Developers Setup Guide** for OSCAR that provides good manual installation instructions. They are supplemented by what is available in the **OSCAR Manual**.

## 3.2 THE CPT SERVICE

The CPT webservice was developed by Leverage for RKL to support a Nova Scotia-centric feature analagos to the service codes used in British Columbia for billing. This service is built on the Spring framework and is implemented in Java. The service provides a simple read only RESTful interface to a basic database of CPT codes that consists of a service table and subset table with a single foreign key relationship indicating the service's membership in a given subset. The data for this table was taken from Dennis Lee's gastroenterology subset. This subset is in the resources directory while a MySQL dump of the database is in the mysql database sub-directory. An application.properties file in the resources directory is provided to configure the application. This is a default configuration directory for Spring applications. Also included is a deployment script which can be run to deploy this service directly from a git clone to the tomcat server.

The application itself has a standard structure with Controller and Model components. These are connected to the CPT database by a package of data access objects (DAOs). This architecture is illustrated in a Unified Modelling Language (UML) diagram in Fig. 1.



**FIGURE 1: THE ARCHITECTURE OF THE CPT SERVICE INCLUDING AND ENTITY RELATIONSHIP (ER) DIAGRAM REPRESENTING THE DATABASE STRUCTURE.**

### 3.3 THE CPT SERVICE API

When employing the CPT Service in an application, each call to the service would require a series of commands to create, send and consume a web request. Through abstraction, this code can be modularized to prevent code clones, unnecessary code replication which degrades software maintainability. For this reason, the CPT Service API was created to provide a module which could be used specifically in Java applications which would rely on the CPT Service. The implementation is still naïve and replicates the model code used by the CPT Service. This code should be extracted into its own package and included as a dependency in each of the CPT Service and CPT Service API modules. Further, work could be invested in developing an executable from the CPT Service API to test it against a CPT Service deployment. This requires some investment in understanding maven and the construction of executable java archives (JARS) and the linking, internally or externally, to dependencies. During prototype implementation in this project challenges with Maven linking resulted in developing the integration into OSCAR with raw code rather than by packaging the CPT Service API as a JAR and including it as a dependency in OSCAR's Maven (OSCAR's build tool) pom.xml file (the Maven configuration file). The updated code is available from the RKL [Github site](#). The architecture of this component is illustrated in Fig. 2.

## 4 OSCAR2 AT CANADIAN WEBHOSTING

Canadian Webhosting (CWH) is a webhosting service employed by RKL. The client management portal to this service is available [here](#). The host used for the OSCAR Clinical Coding project was OSCAR2, a Ubuntu 14.04 LTS system. Dennis Lee is the longitudinal system administrator for this system. Leverage Analytics has been charged with its maintenance during the course of this project. Three primary deliverables have been provided at CWH:

1. A deployment of the current OSCAR 15 BC Beta (Subsection [4.1](#))
2. The Clinical Coding prototype (Subsection [4.2](#))(Subsection [4.3](#))
3. Inclusion of Dennis Lee's EForms in these two deployments

### 4.1 AN OSAR 15 BC BETA DEPLOYMENT

A recent version of OSCAR15 configured for BC has been deployed on the CWH OSCAR2 server. It can be accessed at [OSCAR15\\_BETA](#) with standard credentials available from Fran-

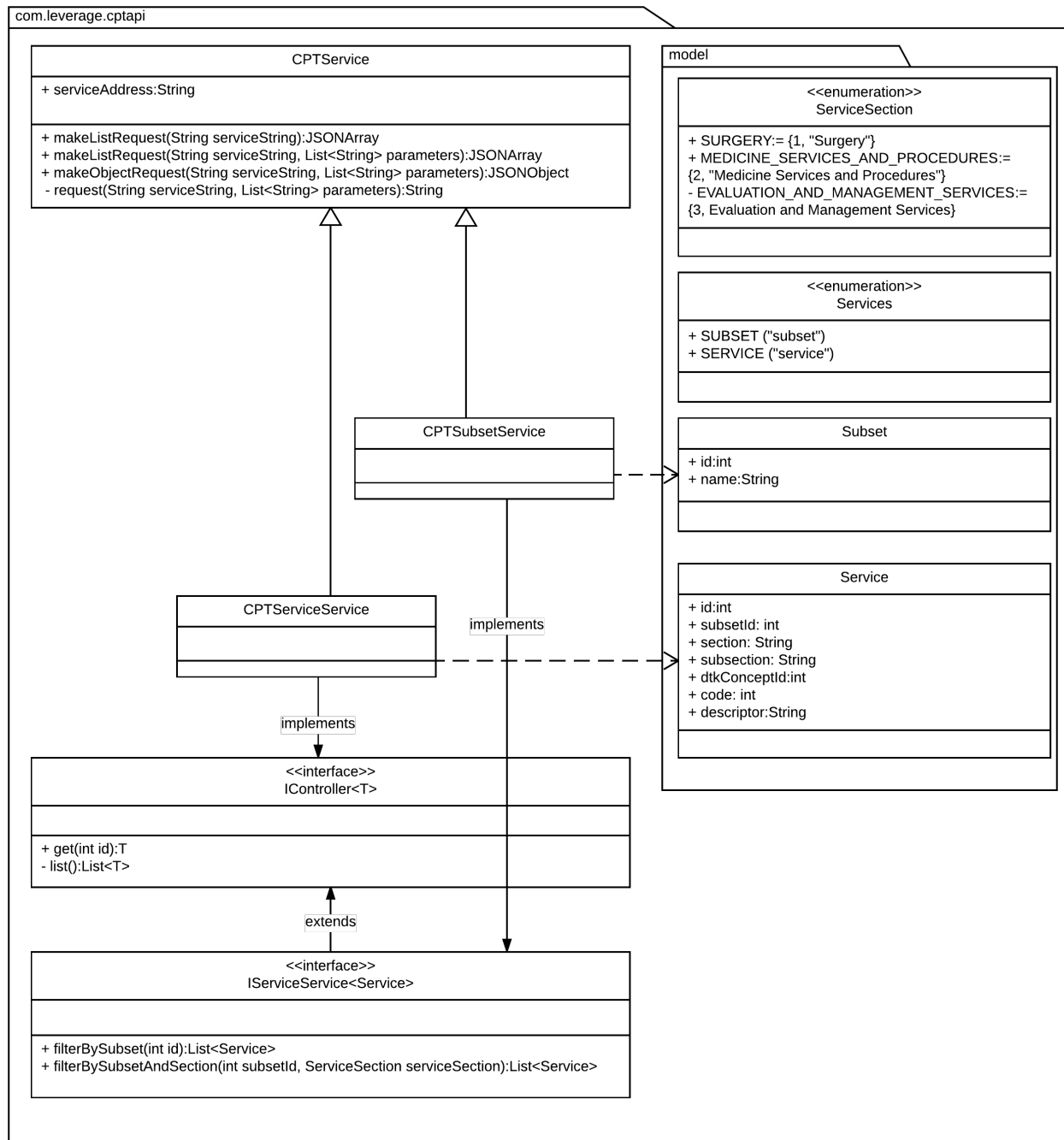


FIGURE 2: THE ARCHITECTURE OF THE CPTSERVICEAPI

cis Lau. This deployment includes the CDA feature previously implemented by Leverage and finally merged into OSCAR Master during the PMMP project.

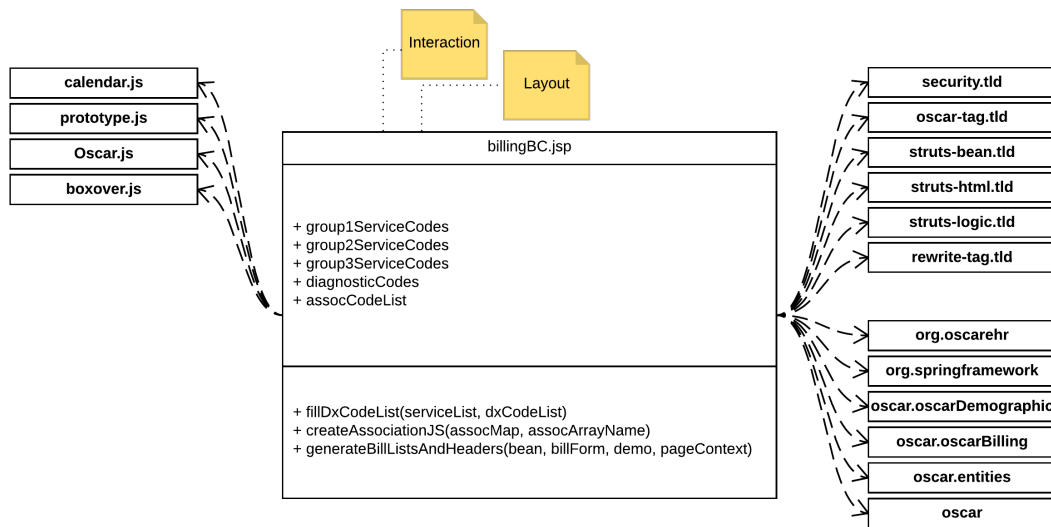
## 4.2 THE CLINICAL CODING A PROTOTYPE

The final technical deliverable provided by Leverage is the Clinical Coding A prototype. This prototype integrates the CPTService, CPTService API and Dennis Lee's SNOMED CT terminology service. The prototype provides a billing interface which, when the deployment has a Nova Scotia configuration, offers CPT codes drawn from Dennis Lee's Gastroenterology subset as service codes and SNOMED CT codes as diagnostic codes. These codes are then used in the creation of a bill for the demographic with which the encounter is associated. This feature, as far as is described is fully integrated in OSCAR. Further effort proved challenging and these issues are further discussed in Section 6. Before addressing these issues however, we provide an review of the effort invested in developing the Clinical Coding A prototype.

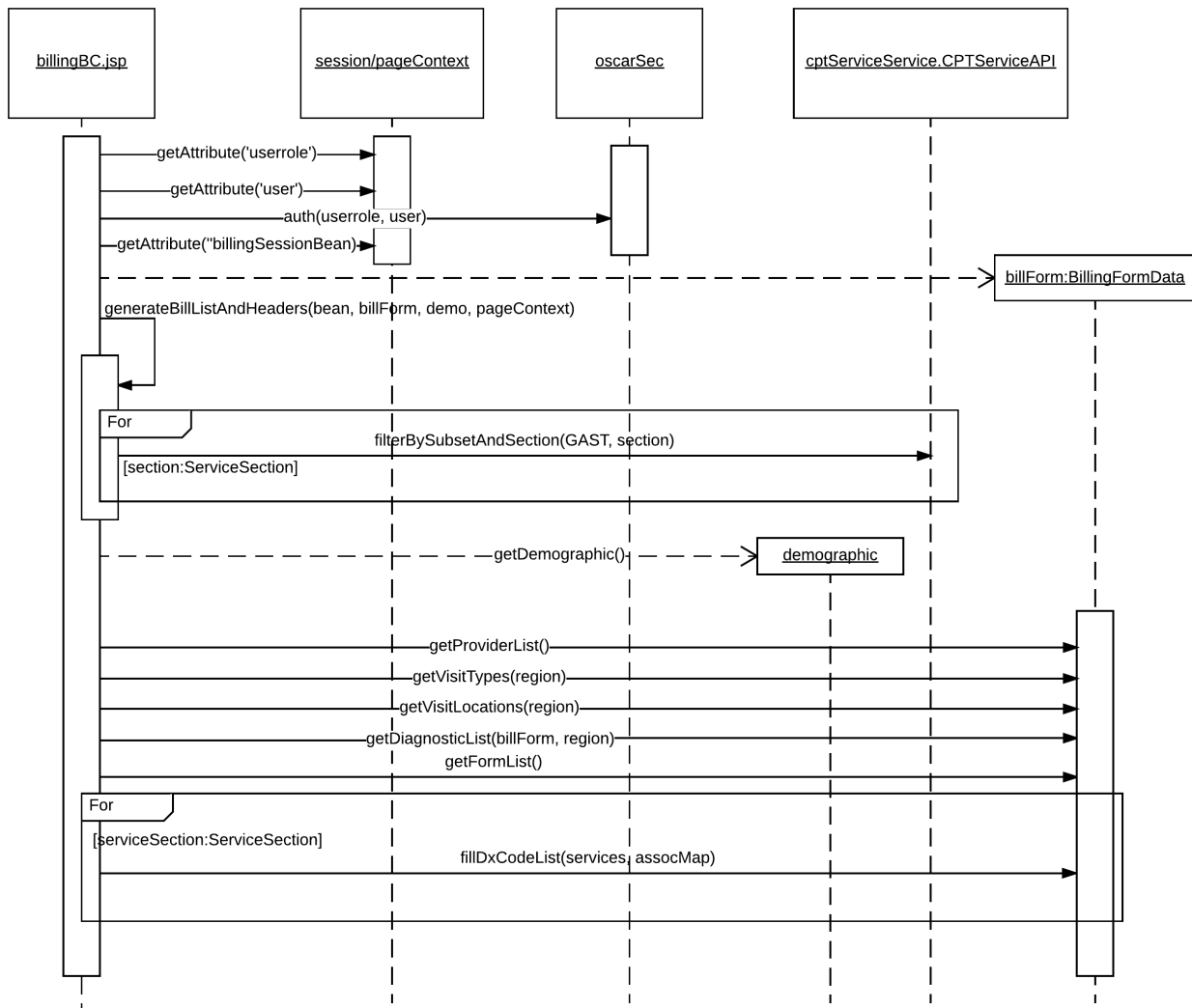
A diagram illustrating the architecture of the BC Billing page is provided in Fig. 3. In Fig. 4 we provide a sequence diagram that illustrates the initialization of the billing page. The layouts of the bill creation form and the bill are shown in Figs. 5 and 6.

The essential effort in developing this prototype was six-fold. First, was the integration of Dennis Lee's SnomedCT terminology server via an autosuggestion feature in the BillingBC.jsp page. This feature required the introduction of a configuration parameter stored in the OSCAR properties file. Second, the BillingBC.jsp page badly needed refactoring. The BillingBC.jsp file in the master branch of OSCAR is chaotic, has not seen maintenance and does not leverage the java standard tag library as has become the norm in OSCAR. These two steps were taken on in tandem as the integration of Dennis Lee's terminology server depended on the Javascript (JS) portion of the OSCAR technology stack. The CPT coding service needed to be constructed to provide replacement service codes for the BillingBC.jsp page. This required our third and fourth efforts being the creation of a new CPT database from Dennis Lee's Gastroenterology CPT subset and the development of a Spring based RESTful service to interface for the database. The fifth major effort was to develop a Java based module to interface with the new CPT service. Unlike Dennis Lee's service, the CPT service was replacing core database functionality in OSCAR and so the integration of this component of the system was more involved. The sixth major effort was to integrate this new module into OSCAR allowing for the use of the CPT service for populating the BillingBC.jsp service codes.

The prototype can be accessed on the web at [Clinical Coding A](#) with standard credentials available from Francis Lau. For administration access, login to the CWH OSCAR2 server is re-



**FIGURE 3: AN ARCHITECTURE DIAGRAM OF THE BC BILLING PAGE WHICH WAS LEVERAGED TO CREATE THE CLINICAL CODING A PROTOTYPE**



**FIGURE 4: A SEQUENCE DIAGRAM REPRESENTING THE INITIALIZATION OF THE BILLING PAGE**

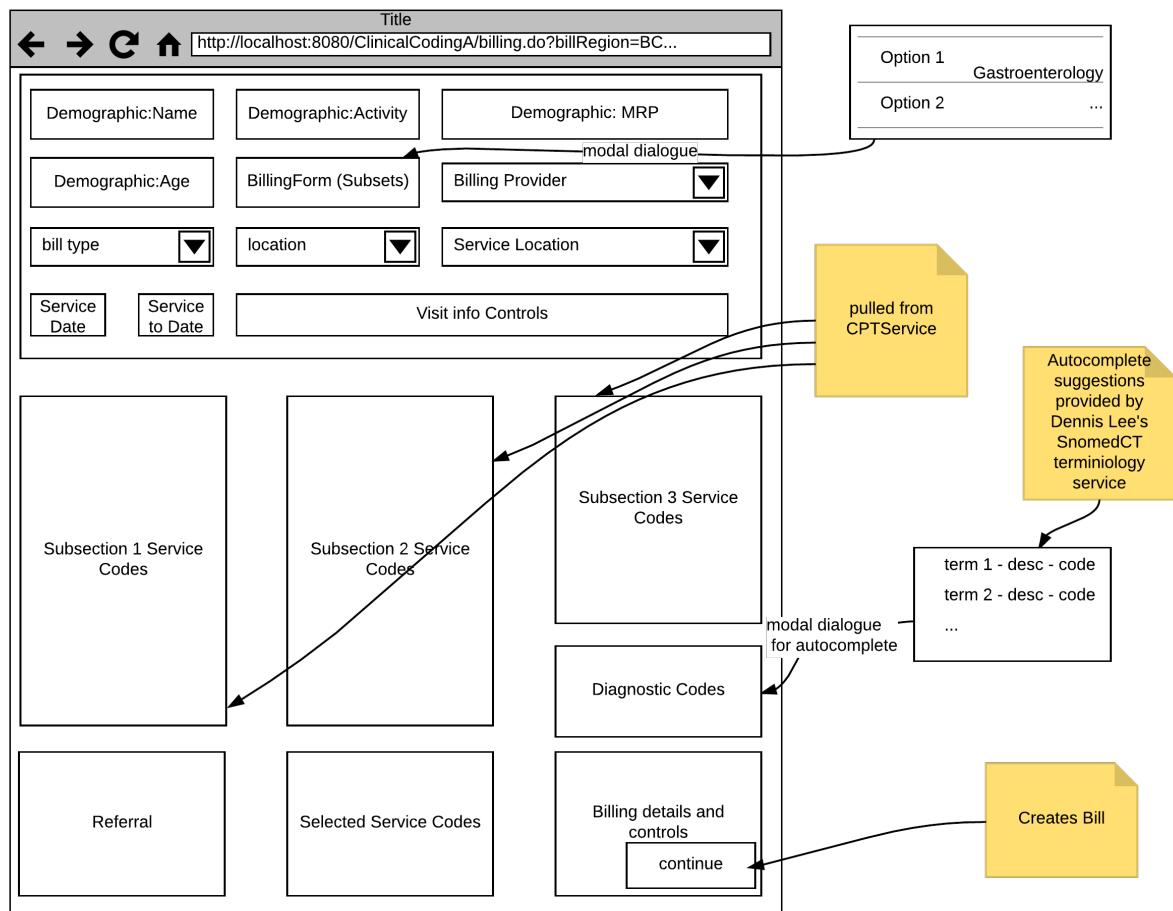


FIGURE 5: A WIREFRAME OF BILLINGBC.JSP



The wireframe shows a browser window with the title 'Title' and the URL 'http://localhost:8080/ClinicalCodingA/billing/CA/BC/CreateBilling.do'. The page contains five form sections stacked vertically: 'Demographics', 'Billing Information', 'Service Code and Description', 'Diagnostic Code and Description', and 'Details and Controls'. A yellow note with a black arrow points to the 'Diagnostic Code and Description' section, stating: 'Not completed as it required the creation of an additional API for Dennis Lee's Snomed Service'.

FIGURE 6: A WIREFRAME OF THE BILLING PAGE

quired and Dennis Lee has this authority.

### 4.3 THE DEPLOYMENT OF DENNIS LEE'S EFORM

Dennis Lee's EOL EForm has been deployed in both delivered OSCAR deployments. Loading of the eforms requires some manipulation of the security privileges for the oscar doc user but is otherwise trivial.

## 5 DEVELOPMENT CHALLENGES AND SOLUTIONS

In this section we will provide tips on working with OSCAR.

### 5.1 MAINTENANCE TIPS

#### 5.1.1 LOCKED OUT?

```
mysql> update security set b_LocalLockSet=0 where user_name='oscardoc';
```

### 5.1.2 WANT TO STOP ACCOUNT EXPIRY?

```
mysql> update security set b_ExpireSet=0 where user_name='oscardoc';
```

## 5.2 DEVELOPMENT TIPS

### 5.2.1 GERRIT PATCH INSTRUCTIONS

credit: Marc Dumontier @ OSCAR EMR

1. login to gerrit [oscartools]
2. go to your commit
3. copy the download link in the top right for checkout - git fetch `ssh://username@source.oscartools.org:29418/oscarrefs/changes/path/to/commit` && git checkout FETCH\_HEAD
4. clone the repository, or if you have one already with no unsaved work, that's fine too
5. cd into the clone
6. git fetch `ssh://username@source.oscartools.org:29418/oscarrefs/changes/path/to/commit` && git checkout FETCH\_HEAD
7. git rebase -i origin/master
8. this will finish, or likely not, and you will have conflicts
9. git status
  - (a) the last set of files are the ones with conflicts..so for example git add `src/main/resources/oscar_mcmaster.properties`
    - do this after editing the file and fixing the conflict
10. if you do that for all the files, and do a git status again, you shouldn't have anything left in that section
11. git rebase -continue
12. should say it successfully rebases it
13. then push to refs for code review again git push `ssh://username@source.oscartools.org:29418/oscar HEAD:refs/changes/path/to/change`

## 5.2.2 BUILDING OSCAR WITH AND WITHOUT UNIT TESTS

Marc Dumontier informs OSCAR Devel on 2012-03-02 21:58:04 ([message #28922315](#)) That if you want to build the project and skip the tests run maven with

- `-Dmaven.test.skip=true`

and that the tests will need

- the mysql user 'oscar' (with password 'oscar') to have full privileges to oscar\_test database to run

If you want to override username, password, db name, just change the variables in `src/test/resources/over_ride_config.properties`.

## 5.2.3 SOLVING SQL GRAMMAR EXCEPTION

This is likely a race condition with unit tests. Turn the hibernate.show\_sql flag in `oscar_mcmaster.properties` on for more verbose output from the build.

- `# true : enable showing sql statement #false : disable showing sql statement`  
`hibernate.show_sql=true`

What is likely required is that you supplement unit test preconditions with the creation of the missing tables identified through the use of the hibernate.show\_sql configuration.

## 5.2.4 ACCEPTING SELF-SIGNED CERTS

Many OSCAR resources use self signed ssl certificates. By default, git will reject these, to allow them:

- `git -c http.sslVerify=false clone https://domain.com/path/to/git`

## 5.2.5 WORKING WITH SUREFIRE

Surefire is one of the maven plugins used by OSCAR. It is a tool that generates error reports for builds which are viewable via an html rendered error log. With particularly fatal build breaks this can be undesirable. The pages can be viewed as they are rendered by exploring the target directory of the build, but if realtime access to the reports is desired, it is preferable if the logs are output to the console rather than used to construct the html report. To do this, use the surefire usefile flag:

- -Dsurefire.useFile=false

Note that in the standard maven output, test assertion failures are reported as "Failures" while unexpected exceptions are reported as "Errors".

### 5.2.6 SPRING APPLICATIONS ON UBUNTU 12.04 LTS ARE A PROBLEM

If you try and deploy a Spring application on Ubuntu 12.04 LTS you may encounter the following error:

- INFO:  
validateJarFile(/var/lib/tomcat7/webapps/CPTService/WEB-INF/lib/tomcat-embed-el-7.0.52.jar) - jar not loaded. See Servlet Spec 2.3, section 9.7.2. Offending class: javax/el/Expression.class

This is likely caused by a tomcat library version incompatibility. To fix this issue you need to upgrade tomcat. v7.0.52 was functional but v7.0.42 was not. Fixing this problem was non-trivial on Ubuntu 12.04 LTS as the standard repositories don't link to the more recent tomcat version. Upgrading the OS to 14.04 and reinstalling tomcat solved the issue.

### 5.2.7 WHAT VERSION OF UBUNTU AM I RUNNING?

- lsb\_release -a

### 5.2.8 WHAT VERSION OF JAVA IS TOMCAT RUNNING

Tomcat runs as a service, usually with an artificial process user. That means that JAVA\_HOME for any real user is irrelevant when answering this question. What matters is what is in the tomcat startup script. To identify the version for a running instance of tomcat try this:

- ps -ef | grep tomcat7

This problem arose when working with Spring and no Spring banner appeared in the logs. Ensuring that tomcat was running on Java 8 solved the problem.

### 5.2.9 LEVERAGING A MYSQL CONNECTOR IN A WAR

If you are building a WAR that relies on a MySQL connector, you may run into a Classpath problem by which Tomcat can not identify the mysql library. This may be solved by adding simlinks to the tomcat home lib directory. It is likely populated with a list of simlinks which point to /usr/share/java. To add these:

1. cd CATALINA\_HOME/lib
2. ln -s ../../java/mysql-connector-java-versionnumber.jar  
mysql-connector-java-versionnumnber.jar

Note that providing an absolute path appears to create a physical rather than symbolic link. This is inferred from the coloring of the resulting listing.

## 6 RECOMMENDATIONS AND CONCLUSIONS

---

### 6.1 RECOMMENDATIONS

If RKL wishes to start this investigation again, a series of recommendations are provided below

1. If project budgets do not support the extension of OSCAR to support a Nova Scotia jurisdiction, then fully integrated features should not be attempted.
2. If fully integrated features are not implemented, isolated functionality should be developed instead using OSCAR interfaces like EForms, or by developing connected applications using the OSCAR web services (ws) packages.
3. If isolated functionality is to be developed using OSCAR, appropriate talent should be sourced for the work. Applicants with the following skill set should be sought:
  - (a) Java (Advanced Experience 5+ years)
  - (b) Java Spring (Moderate Experience 2+ years)
  - (c) Java Struts (Moderate Experience 2+ years)
  - (d) Tomcat (Advanced Experience 5+ years)
  - (e) Linux System Administration (Moderate Experience 5+ years)
  - (f) Relational Databases (Moderate Experience 5+ years)
  - (g) Hibernate (Beginner Experience)
  - (h) Java Persistence API (JPA) (Beginner Experience)
  - (i) Java Script (Beginner Experience)
  - (j) Angular JS (Beginner Experience)
  - (k) Git (Moderate experience 2+ years)

- (l) Gerrit/Jenkins (Moderate Experience 2+ years)
- (m) HL7 CDA experience (Moderate Experience 2+ years)
- (n) Software Development (Advanced Experience 10+ years)

## 6.2 CONCLUSIONS

On account of development difficulties including issues with the quality of the OSCAR software architecture, the success of the OSCAR Clinical Coding project was limited. In particular, the billing modules in OSCAR which are jurisdiction specific are not well modularized. This results in significant challenges in establishing sandboxed environments in which to develop clinical coding features in this section of OSCAR. Developing in an existing module introduces different problems which compound the existing architecture issues in this EMR.

What has been accomplished is the development of a CPT service which could be used in any clinical coding feature regardless of product vendor. Significant knowledge of OSCAR and its technologies have also been reinforced, and some of this knowledge is expressed in this document and through the various technical deliverables which have been provided to RKL. RKL has also been left with two functional OSCAR deployments, one which uses the latest code from the OSCAR15 BETA release and is configured for the BC jurisdiction, and one which is a little behind this release, but includes the clinical coding prototype work and is configured for a Nova Scotia jurisdiction - currently an extension of the BC jurisdiction on account of the architecture challenges which we have discussed.