Implementing Novel Features to Improve De-duplication User Experience

Primary mentor	Unknown User (surangak)
Backup mentor	Unknown User (sgrannis) Unknown User (judy)
Assigned to	Unknown User (pulasthi7)

Introduction

Duplicate patient records often arise in electronic medical record systems. These duplicates cause fragmentation of a patient records and hinder access to seamless integrated patient data. The PatientMatching module is a tool that helps OpenMRS installations to identify and merge duplicate Patient records arising within the OpenMRS database. The PatientMatching module has been incrementally developed over the last few years by a cohort of Google Summer of Code interns, systems engineers, and Medical Informatics Researchers. Our hope is that GSoC 2012 will see continued success in evolving the module's functionality.

Objectives

This objective of the 2012 GSoC de-duplication module project is to improve the user experience with the de-duplication workflow. The GSoC applicant will help to analyze, design, develop, and implement a number of prioritized features, which include:

Task 1: Incorporate a process to validate de-duplication strategies.
Configuring a de-duplication strategy to find potential duplicates is a moderately complex task. If configured incorrectly, the de-duplication process may fail, or the linkage results may be inaccurate. To ensure a properly configured linkage approach, we will incorporate a validation process that highlights errors in the linkage configuration. Examples of invalid matching configurations include undefined blocking field(s), undefined matching fields, etc.

Task 2: Incorporate a process to calculate total number of potential pairs formed by particular blocking strategy.
The de-duplication module searches through "record pairs" that have a high likelihood for being duplicates. "Records pairs" are formed using "blocking strategies", which are simple approaches to finding similar records by requiring that one or more corresponding fields exactly agree among 2 records. For example, we may stipulate that records agreeing on last and first name should be used to create potential pairs. Occasionally, however, the user may choose a blocking strategy that results in very large (e.g., hundreds of millions, if not billions of record pairs), or very low (zero to less than a hundred) numbers of record pairs. Widely varying numbers of record pairs can result in unexpected results, including out-of-memory errors, excessively long runtimes, confusing or inaccurate results, etc. To avoid this situation, the GSoC intern will implement a function that calculates the estimated number of pairs to be formed, and alerts the end-user to that number. The end user will then have the option of canceling or editing the particular de-duplication strategy.

Task 3: Upgrade the de-duplication reports from flat files to database persistence.
The de-duplication module creates reports listing potentially duplicate records, which end-users can manually review and merge when necessary. Until recently, these "de-duplication reports" were stored as flat files. Unfortunately, flat files limit our ability to manage the data and hinder new creative ways to display the data. Therefore, upgrading from flat files to persisting the data in a relational database will help users and developers more meaningfully use this data. The successful applicant will continue the previously initiated work for this task. For a detailed description of what we have completed so far, and for more hints on how to complete this ticket, see here.

Task 4: Implement a process to analyze and highlight useful de-duplications fields
Data fields in OpenMRS often *appear* to be useful for de-duplication, but are not. This can be the case for a variety of reasons: data may be incompletely or inaccurately recorded, some fields may simply lack the discriminating power to be meaningfully used as matching variables, etc. To rapidly identify data fields that optimally support de-duplication, we've developed data quality and information content metrics that characterize the usability of fields specifically for use with de-duplication. This information can help guide the de-duplication user when selecting specific fields for duplication strategies.

Task 5: Implement additional duplication features in the OpenMRS de-duplication module.
The OpenMRS PatientMatching module also implements a standalone record matching and de-duplication application called "RecMatch." Written as a Java Swing application, "RecMatch" offers expanded de-duplication features and functionalities beyond what currently exists in the de-duplication module. We aim to incorporate a subset of these functions in the de-duplication module's web interface.

Expectations

We understand that the student may have a hard time getting up to speed on some of these tasks. Therefore, they will start with well-scoped, lower-complexity tasks. The student will move onto more complex tasks after achieving an acceptable level of proficiency, and will ideally complete additional tasks as time permits.

Additional credit

Task 6: Migrate previous reports from the flat file to the database
Once Task 3 is completed, all reports will be persisted in the system database. However implementations already using the de-duplication module may have older reports currently persisted as flat files. In order to prevent further ambiguity, the PatientMatching module should provide a run-once operation that imports the flat file and persists older reports into the database as well. This operation should be incorporated into the latest PatientMatching release that contains the database changes. When the user installs the PatienMatching omod it will search for previous "flat-file" reports and move them into the database, rendering the flat file obsolete. This ensures that older report data can also be managed efficiently via the database.

Applicant skills

1. Applicant should be familiar with Apache maven, Spring framework, Hibernate and JSP
2. Applicant should be very proficient in java
3. Applicant is expected to spend spend some time understanding how the PatientMatching module works.
4. The initial ticket for task number 3, PTM-47

Resources

1. You can checkout the latest PatientMatching code from here
2. For a more detailed description of how PatientMatching works, study wiki pages No.1 and 2

3. A Screencast on the PatientMatching module is also available here
4. Previous GSoC project pages can be found here: GSOC 2011, GSoC 2010 and GSoC 2009.

4. Feel free to contact us at surangakas at gmail dot com or sgrannis at regenstrief dot org for further clarifications