RAPTOR Contraindication Engine Architecture

Version 0.88

DRAFT 20131117

Contents

[Introduction 2](#_Toc372494622)

[High Level Goals of the Contraindication Engine 2](#_Toc372494623)

[Limitations and Constraints of the Contraindication Engine 2](#_Toc372494624)

[Assumptions 3](#_Toc372494625)

[Approach Overview 3](#_Toc372494626)

[Flags and Measures 4](#_Toc372494627)

[Available Measures 4](#_Toc372494628)

[Defining Flags 5](#_Toc372494629)

[Available Baseline Flags 6](#_Toc372494630)

[Contraindication Rules 6](#_Toc372494631)

[Trigger Criteria 7](#_Toc372494632)

[How Null Values Impact Rule Evaluation 7](#_Toc372494633)

[Compiler Features 7](#_Toc372494634)

[Deployment 8](#_Toc372494635)

[Revision History 9](#_Toc372494636)

[Table 1 - Rule-base modification workflow 3](#_Toc372494637)

[Table 2 - Runtime rule evaluation process 3](#_Toc372494638)

[Table 3 - Flags and Measures 4](#_Toc372494639)

[Table 4 - Available Measures 4](#_Toc372494640)

[Table 5 - How system gets a measure value at runtime 4](#_Toc372494641)

[Table 6 - Flag types 5](#_Toc372494642)

[Table 7 - Phrase Match Criteria Sections 5](#_Toc372494643)

[Table 8 - Baseline Flags 6](#_Toc372494644)

[Table 9 - Rule Attributes 6](#_Toc372494645)

[Table 10 - Trigger Criteria Sections 7](#_Toc372494646)

# Introduction

As part of the RAPTOR functionality, medical professionals are warned of some potential contraindication conditions when they are detected by the application. The set of contraindications detected by the application is configurable by the hospital center.

NOTE: Details of this design are subject to changes if underlying assumptions about available data values in the runtime environment are changed.

# High Level Goals of the Contraindication Engine

The contraindication engine is architected to meet all the following goals.

1. Contraindications are detected and reported while user is engaged in RAPTOR workflow process
2. A baseline set of contraindication detections are delivered with the core RAPTOR application
3. RAPTOR will have a mechanism for reporting detected contraindications and requiring acknowledgement of the contraindication by user of the system.
4. Hospital centers can add additional contraindication detections to RAPTOR for their particular installation.
5. Hospital centers can edit any existing contraindication detections in RAPTOR for their particular installation.
6. Hospital centers can customize the warning message text reported by RAPTOR when a contraindication is detected.
7. Appropriately privileged users of the RAPTOR system can audit the deployed contraindication rule-base.

# Limitations and Constraints of the Contraindication Engine

The contraindication engine will have the following limitations.

1. RAPTOR will only detect contraindications that have been coded into its deployed rule-base
2. The contraindication rules are global for all the users of a RAPTOR installation.
3. Contraindication detection rules that depend on phrase matches can be circumvented with alternate spellings and misspellings or abbreviations in the source material that is being scanned by the engine.
4. Assumption that a phrase match indicates a relevant condition can be derailed by unexpected phrasing. (e.g., phrase “is not obese” would not indicate obesity condition although a simple word match for single word phrase “obese” would produce a match.)
5. Contraindication detection rules that depend on specific numeric values can be circumvented by values that have not been entered in the expected format or are not entered using expected units.
6. The engine has a limited number of input sources which it scans for information, if the relevant match criteria is not in those sources the engine will have no way of detecting the contraindication.
7. Detected contraindications are only reported within the RAPTOR application, not to other systems.
8. RAPTOR cannot prevent Illogical or wrong contraindication criteria being provided to it
9. Updating a rule-base will require a system administrator to deploy the updated rule-base

# Assumptions

The design of the contraindication engine and deployed for RAPTOR will be based on the following assumptions.

1. An updated rule-base will only be deployed after appropriate medical and technical scrutiny
2. RAPTOR will not have more than 100 simultaneous users per installation
3. The engine will have no more than 100 contraindication rules
4. The engine will have no more than 100 named measures
5. The engine will have no more than 100 named flags
6. Data sources are available through MDWS and are within scope of the RAPTOR project to retrieve values needed by the contraindication engine
7. All relevant data sources will have stable names that can be referenced from the rules.

# Approach Overview

The contraindication engine is a separately testable module of the RAPTOR application. The workflow for updating and releasing updated collections of contraindications detected by the engine is a separate workflow from that of the RAPTOR application itself.

Table - Rule-base modification workflow

| Step | Participant(s) | Description |
| --- | --- | --- |
| 1 | Medical SME | Create/Edit/Remove flags and contraindication rules |
| 2 | Medical SME and System Administrator | Compile the flags and rules into a deployable rule-base |
| 3 | System Administrator | Deploy the new rule-base |

The internal process of the contraindication engine at runtime operates as illustrated in Table 2. That process repeats every time there is a workflow state change in RAPTOR on a ticket that has not yet been terminated.

Table - Runtime rule evaluation process

| Step | Phase | Description |
| --- | --- | --- |
| 1 | Prepare for evaluations | Examine all available information sources to set flags and measures so they are available to contraindication rules. |
| 2 | Evaluate rules | Evaluate all the rules to detect contraindications relevant to the current set of data. |
| 3 | Report contraindications | Report to the RAPTOR application all the contraindication rules for those rules that detected a contraindication. |

## Flags and Measures

The contraindication engine has a set of capabilities to produce flags and measures prior to evaluation of rules. These flags and measures can be thought of as key values which rules are then checking to determine if a contraindication has been triggered.

Table - Flags and Measures

| Term | Description |
| --- | --- |
| Flags | These are named items that have one of the following three values   1. True 2. False 3. Unknown (null) |
| Measures | These are named items that have one of the following two values   1. A real number value 2. Unknown (null) |

### Available Measures

There are a finite set of measures available for use in rules and flag criteria. See Table 4 for the complete list.

Table - Available Measures

| Name | Data Sources | Description |
| --- | --- | --- |
| MIN\_EGFR\_10DAYS | TBD | Minimum eGFR value within the most recent 10 days |
| MIN\_EGFR\_15DAYS | TBD | Minimum eGFR value within the most recent 15 days |
| MIN\_EGFR\_30DAYS | TBD | Minimum eGFR value within the most recent 30 days |
| MIN\_EGFR\_45DAYS | TBD | Minimum eGFR value within the most recent 45 days |
| MIN\_EGFR\_60DAYS | TBD | Minimum eGFR value within the most recent 60 days |
| MIN\_EGFR\_90DAYS | TBD | Minimum eGFR value within the most recent 90 days |
| RECENT\_WEIGHT\_KG | TBD | Most recent available patient weight (kg) |
| RECENT\_WEIGHT\_LBS | TBD | Most recent available patient weight (US pounds) |
| CURRENT\_AGE | TBD | Current age of the patient in years |

The value of a named measure will depend on the available data at time of evaluation. See Table 5 for conditions under which you can expect a real number value vs. having a Null result.

Table - How system gets a measure value at runtime

| Value | Description |
| --- | --- |
| Null result | Null result if any of the following apply:   * + the data sources are empty   + data sources are not empty but number match could not be found   + data sources are not empty but more than one number match was found |
| Real Number | One real number was found by the system. |

### Defining Flags

The collection of available flags is entirely configurable. The types of flags are as follows:

Table - Flag types

| Flag Type | Attributes | Description |
| --- | --- | --- |
| Phrase Match | * Flag Name * Data Sources * Match Criteria | * Flag is set to true or false depending on the Match Criteria evaluation as explained in Table 7. |
| Numeric Formula | * Flag Name * Data Sources * Reference Value * Relation Type * Formula | * Flag is set to Unknown (Null) if any of the following apply:   + the data sources are empty   + data sources are not empty but number could not be found   + data sources are not empty but more than one number match was found * If Relation Type is EQUAL then value will be true only if formula evaluates to the same value as the reference value. * If Relation Type is LESSTHANOREQUAL then value will be true only if formula evaluates to a value less than or equal to the reference value. * If Relation Type is LESSTHAN then value will be true only if formula evaluates to a value less than the reference value. * If Relation Type is MORETHANOREQUAL then value will be true only if formula evaluates to a value more than or equal to the reference value. * If Relation Type is MORETHAN then value will be true only if formula evaluates to a value more than the reference value. |

#### Sections of the Phrase Match Flag Type’s Match Criteria Attribute

So that the flag does more than simply match a list of phrases, the “Match Criteria” attribute contains the sections describe in Table 7. The match criteria must have at least one of the sections filled, and the flag is set to true only if all the non-empty sections evaluate to true. If any filled section evaluates to false, then the entire flag evaluates to false. Empty sections are simply ignored for the evaluation of the flag’s value.

NOTE: A phrase is simply a short collection of text that may contain one or more words. Letter case is ignored and all whitespace is normalized into a single space for all phrase matching.

Table - Phrase Match Criteria Sections

| Section Name | Description |
| --- | --- |
| At Least One Phrase Matches | Section is True only if at least one phrase from this list matches phrases in the data sources. |
| All Phrases Match | Section is True only if all phrases listed here match phrases in the data sources. |
| No Phrases Match | Section is True only if none of the phrases listed here match phrases in the data sources. |

### Available Baseline Flags

The following flags will be available in the baseline engine installation and can be customized by the administrator of the engine.

Table - Baseline Flags

| Name | Data Sources | Description |
| --- | --- | --- |
| IS\_DIAGNOSTIC\_EXAM | TBD | True if this is a diagnostic examination |
| IS\_IMG\_GUIDED\_EXAM | TBD | True if this is an image guided exam |
| IS\_POSSIBLE\_DUP\_PROC | TBD | True if this is potentially a duplicate procedure |
| IS\_RENAL\_IMP\_STG\_4\_OR\_5 | TBD | True if patient is potentially in renal impairment stage 4 or 5 based on eGFR measures. |
| IS\_RENAL\_IMP\_ACUTE\_STG\_3 | TBD | True if patient is potentially in renal impairment acute stage 3 based on eGFR measures. |
| IS\_RENAL\_IMP\_CHRON\_STG\_3 | TBD | True if patient is potentially in renal impairment chronic stage 3 based on eGFR measures. |
| IS\_MALE | TBD | True if patient is male |
| IS\_FEMALE | TBD | True if patient is female |

For maintenance clarity, all flag names must start with the text “IS\_”.

## Contraindication Rules

Each rule is independently created and managed in what we call the “rule-base”. The rule-base is simply the collection of all flags and rules available to the contraindication engine. Each contraindication rule of the rule-base has the fixed set of attributes described in Table 9.

Table - Rule Attributes

| Attribute Name | Description |
| --- | --- |
| Rule Category Name | This value is simply used for grouping results in the display to the user. |
| Rule Name | This is a unique name. Each rule must have a unique name that starts with the text “RULE\_”. |
| Rule Explanation | This is static text that explains to a Medical SME what the purpose of this rule is. This might include references to specifications or common medical practice. This text will not be interpreted in any way by the contraindication engine. |
| Message Text | This is the text of the message that will be shown to the user if the contraindication triggering condition is detected. The message text will contain the specific values found in the data that triggered the message display. |
| Require Acknowledgement | This is either set to True or False. If set to True, then the detected contraindication will require acknowledgement by the user to continue. If set as False, then the message will be displayed but no user acknowledgement will be required. |
| Trigger Criteria | This is the detection criteria that the contraindication engine will evaluate to determine if the contraindication condition applies to a specific set of data. |

### Trigger Criteria

The trigger criteria of a contraindication rule has very little Boolean expression sophistication so that overall complexity of rule definition and validation is minimized. The driver for this simplification is so that each rule can be managed and validated by Medical SMEs in a clear manner.

Instead of general purpose Boolean expression syntax, the trigger criteria for a rule is defined by filling in the sections described in Table 10. At least one section must be filled in and for the rule to be triggered each of the non-empty sections must evaluate to true.

Table - Trigger Criteria Sections

| Criteria Section Name | Description |
| --- | --- |
| All Flags True | Section is True only if all flags listed here evaluate to True. |
| All Flags False | Section is True only if all flags listed here evaluate to False. |
| All Flags Null | Section is True only if all flags listed here evaluate to Null. |
| All Triggered Rules | Section is True only if all rules listed here were already triggered. |
| All Non-Triggered Rules | Section is True only if all rules listed here were not triggered. |

NOTE: To simplify potential dependency chains, a rule on which other triggering criteria depends cannot itself depend on any other rules.

## How Null Values Impact Rule Evaluation

A null value (also describable as an “unknown value”) appearing in a formula will by default make the result of that formula a null value. Null values are not equivalent to real True/False values and are not equivalent to real number values.

When encountering a null value in trigger criteria that did not expect it, the engine will produce a warning to the user that there was insufficient information to evaluate that contraindication trigger.

# Compiler Features

The compiler will examine all candidate rule-base settings to ensure the following is satisfied.

1. All mandatory attributes of a flag have been filled in
2. All mandatory attributes of a rule have been filled in
3. All data sources declared are real sources
4. All flag names are unique and start with text “IS\_”
5. All rule names are unique and start with the text “RULE\_”
6. All names in criteria are valid (name things that are appropriate for the section and exist)
7. No contraindication rule depends on a rule that itself depends on another contraindication rule

A successfully complied rule-base can then be deployed by a RAPTOR system admin with sufficient authorization and privileges on the application server.

# Deployment

We will leverage agile analysis, design and development strategies to resolve the deployment strategy by the summer of 2014. This will likely have to incorporate a testing strategy and policy component.

# Revision History

|  |  |  |
| --- | --- | --- |
| When | Who | What |
| 20131116 | Frank Font | Initial draft |
| 20131117 | Frank Font | Added details for flags and measures and refined assumptions |
|  |  |  |