Functional Requirements/ Statements of Function

All models would be:

Externally developed

Externally Loaded into a target directory

Target Claims Data Sets externally loaded

BCDS auto-ingest the records into the repository

Conformance validation, format verification records will likely be:

Hard-coded, or;

Entered once into a table manually during development

Ultimately the system would be pointed at an isolated ingestion point

BCDS would auto-ingest the selected records

Result Sets would be:

Auto-ejected

Stored in the selected target location

Both the selected target claims data and modeling result sets would be auto-ingested by system and the comparative analysis would be auto-ejected following completion of the run to the selected destination location.

Detailed User interface designs:

Necessary for each user type

data feeds will most likely be coming from the current system of record

In a production sense we will be pulling data

service abstracts the disparate repositories, as well as record sources.

two considerations:

one time pulls for use in developing models

ongoing data feeds

envision test database growing

role of a constant learning system,

tuned as part of its operation

'advised' by one of the users against metric goals (more accuracy, higher throughput, etc.)

envision this being a data feed that would resemble a queue that we register our service on

process of learning

multiple users attempting to do very intense model development

operations would be affected

form an application

model against incoming feeds

this should be low intensity

not on-demand

following a batch (In this case users are passive and will not really affect things)

web based platform

Standard login page

may evolve to have the appropriate graphics and UI needs of the VA.

The view will be tailored to the User's role

various views based on role

navigation maps need to be worked from there

considering the overall user experience.

anticipate model execution to be a service based operation without user interaction

tailoring of the data, "success" thresholds, and insight into actual execution (success, warning, errors, etc.)

operations are agnostic as to the actual model

really a reflection of the abstraction of a model interaction

possible extensions will be necessary for certain models

consider the case of that potential