

Clinical Effeciveness Group

in partnership with

Endeavour Health Charitable Trust

Patient State calculator

A sub project to the ENTERPRISE project

Version 1.1

Draft

**Amendment History**

|  |  |  |
| --- | --- | --- |
| Version | Date | Amendment History |
| 1.0 | 19/1/2015 | Initial draft |
| 1.1 | 24/1/2015 | Changes current limitations and addition of state calculator service |
|  |  |  |

**Authors and Reviewers**

This document must be reviewed by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title / Responsibility | Date | Version |
| David Stables | Trustee , Endeavour Health | 19/1/2015 | 1.0 |
| Dr Kambiz Boomla | SRO – CEG Enhanced data Service |  |  |
|  |  |  |  |

**Approval**

This document must be approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title / Responsibility | Date | Version |
| Dr Kambiz Boomla | SRO – CEG Enhanced data Service |  |  |
| Dr John Robson |  |  |  |

Contents

[1 Document Purpose 4](#_Toc441403078)

[2 Background 4](#_Toc441403079)

[3 Requirements 6](#_Toc441403080)

[3.1 State designer 6](#_Toc441403081)

[3.1.1 Installation and access to state designer 7](#_Toc441403082)

[3.1.2 Multi-user role support 7](#_Toc441403083)

[3.1.3 State manager module 7](#_Toc441403084)

[3.1.4 State definition workflow 7](#_Toc441403085)

[3.2 State Editor 8](#_Toc441403086)

[3.2.1 Query definition 8](#_Toc441403087)

[3.2.2 Assignment of variables and values 8](#_Toc441403088)

[3.2.3 Creation of the formula function 8](#_Toc441403089)

[3.2.4 Selection of clinical code 8](#_Toc441403090)

[3.2.5 Editor Output 8](#_Toc441403091)

[3.2.6 State definition library 8](#_Toc441403092)

[3.3 State calculator 9](#_Toc441403093)

[3.4 State messenger 9](#_Toc441403094)

[3.5 Re-identification Service 10](#_Toc441403095)

[3.6 State calculator Service 10](#_Toc441403096)

# Document Purpose

The purpose of this document is to specify requirements for a set of software components and services for a subproject that is part of the CEG enhanced data access service project (referred to as the ENTERPRISE project).

The area covered by this document includes the generation of risk scores or similar derived scores (referred to as *states)* and the posting of these scores back into the relevant source clinical system such as the GP system

The components and services will calculate *patient state* based on pseudonymised care record data from a number of different source organisations, create a structured message and using the reverse identification mechanism, will post the information to the patient’s GP practice, and record the state in the patient’s electronic health record.

# Background

Before making a decision on care, a care professional assesses the *state* of a patient.

The clinician usually starts by obtaining and collating state information about the patient from a number of sources.

For example, sometimes the source is the patient (“I am short of breath/ tired”), or the clinician (“appears pale”) or a device (“Haemoglobin =6.5”) or another clinician (“film= suggestive of microcytic anaemia”), and sometimes the source is a previously recorded state (“last consultation: Upper abdominal pain relieved by food”).

The next step is for the clinician to infer state, taking account of a pattern of findings (which are themselves states).

For example in the above, the state of “microcytic anaemia” is a reasonable inference with a very high level of confidence. The state of “anaemia due to blood loss” is highly likely (perhaps 95%), and an underlying causal state of “duodenal ulcer” is very likely, although there may be some other cause.

Computers supplement human mediated state inference with computable state inference. Sometimes this inference results from Boolean logic operating on care record entries and sometimes this results from arithmetic formulas operating on variables, themselves derived from record entries.

For example, a “BMI” is a state calculated from a variable “H” which is derived from the latest recorded entry of height in the record, variable “W” which is derived from the latest entry of recorded weight and calculated via the formula “W/H2”.

An example of a Boolean logic calculated state might be “If patient has asthma AND is on dual asthma medication and has NOT had a review for Asthma for the last year then the state might be “Overdue Asthma review”)

States can be said to have a temporal classification of current, past, or predicted. A particular BMI value may be a current state or a past state when they have lost weight. An example of a predicted state is a risk of suffering a cardiovascular event. This uses a formula that operates on variable values derived from a number of state entries in the record derived from a *record query.*

Predicted states may also operate on projected trends of state entry values in the record.

The entry of inferred state into a person’s record is of value as these entries can form the basis of further state inference and can inform care decisions. To put it another way, if a computer can calculate state and subsequently record state, the professional can make better decisions and health might improve. The calculation of state by a computer uses only a tiny amount of electrical energy, at a trivial cost so there is a good cost/ benefit ratio.

Sometimes it is better for a user to confirm a state before entering data into the record. In the above example, with very little user involvement the State of “Active Problem: Microcytic anaemia” could quickly be entered into the record together with the likely cause of blood loss, whereas the underlying cause would not have sufficient confidence.

Currently a healthcare professional can use computable states in a variety of ways including:

1. They can search for patients with certain characteristics, using *record query* as the means of calculating states, the result being a list of patients.
2. They can query the patient record automatically when they are consulting with the patient, the result being an alert (e.g. concepts in EMIS)
3. They can calculate state with the use of a formula. The formula queries the record and using variable values assigned to the query result set generates a score or value (e.g. QRisk)

There are a number of limitations with the current implementations, including:

1. Complex formula based state calculators use proprietary query software and proprietary formulae calculation software, even when the formulas are public domain, and they are often opaque to the user.
2. The demand for state calculation outstrips the supply of state calculators by a large margin.
3. State calculations that operate at the point of care, such as EMIS concepts, use less complex logic than the search engine and do not support complex formulas.
4. Currently if a protocol informs the GP of a new state the patient has acquired, perhaps through a protocol driven alert, there is no way of knowing whether the protocol has been run or the alert seen. It is consequently difficult to evaluate the effectiveness of the alert in changing behaviour.

An example is an alert that pops up to say the patient is “using a lot of salbutamol and is not on an inhaled steroid”.  The question then arises: “Does this alert change behaviour?” Unless there is a log in the patient record that the alert has run, then you cannot know.

1. Systems often operate only on data held in a particular system for a particular organisation, i.e. they miss many record sources when inferring state.

The software components specified in this document address the above limitations in the following ways:

1. An editor tool will allow informaticians to specify complex calculators in order to remove the reliance on proprietary software.
2. The number of calculators will be limited to the available knowledge rather than the available software. Local policy will determine which ones are used.
3. Calculators operating at the point of care will be able to use the same level of sophistication as the search and reporting models and formulas combined
4. The service can generate the state values and store them in the record.
5. The informaticians will be able to run the algorithms against an entire population and operate on multiple data sources concurrently.

# Requirements

Four components are required. These are:

1. **A State designer**

A state designer is an application that enables clinicians to define rules and formulas based on knowledge generated from research studies on the population data store or from research papers published from other studies.

1. **State calculator**

The state calculator is a software component that runs on the care record and generates state. It can operate within the data service on a population or within the data service on a particular patient, or in the consultation application on the local care record.

1. **State messenger service**

The state messenger service is a set of components within the data service that can receive state information generated by the state calculator and transmit a message containing the state value to the provider organisation’s clinical record or to a workflow application.

The state messenger service will use the RESOLUTION integration service for this purpose.

1. **Re-identification service**

The re-identification service is a secure service that uses the forward hash link table within the CSU to re-match the pseudonymised patient identifier to the source patient identifier to be able to post back the data to the appropriate patient and source organisation.

1. **State calculator service**

This is an externally facing Web service or API that enables a client clinical system to request a score in real time based on the contents of a current consultation and the contents of the patients EHR (which also contains information from other sources)

## State designer

The state designer is an application that enables clinicians with informatics skills and some basic software skills (or access to developers with basic programming skills), to define rules and formulas based on knowledge.

The knowledge is generated either from research studies on the local population data store, or from research papers published from other studies operating on other population data stores.

### Installation and access to state designer

The state designer will be available in a test environment as an open source application via the Endeavour web site.

A distribution manger is responsible for vetting and accepting requests to install and use the state designer for the live system. The distribution manager will supply access keys and initial install and log on credentials.

It should be possible for a user with the relevant access key to install and log on to the live system. On initial log on the user will have the role of “reviewer” unless already configured my the system manager (see below)

### Multi-user role support

The state designer application should support at least three role levels, which are:

1. System manager role
2. Author role
3. Reviewer role

A system manager configures and upgrades user roles.

An author can create and edit calculators and is able to assign state definitions to release status and assign versions.

A reviewer can review all state definitions and can create draft state definitions but cannot assign release status.

### State manager module

The state manager module is responsible for

1. A view of the state definitions, including current and previous versions when required
2. Managing files and folders
3. Setting the status of state definitions as being draft or release status

It should be possible for an author to configure folders, update the status of a definition and update the version of the definition.

It should be possible for an author to review previous versions of the definition and revert a definition to a previous version.

It should be possible for an author or a reviewer to review the definitions and mark the definition as viewed.

### State definition workflow

A user should have access to an inbox containing a list of state definitions that they may wish to work with.

There should be a facility to assign a state definition to one or more user’s inbox.

All actions undertaken on the item should be consistent with the user’s role and the workflow manager should support all actions.

## State Editor

A user should be able to create edit, delete and modify a state definition.

### Query definition

A state editor should support complex multi-rule query to the full extent of the query editor’s capability in order to obtain result sets, the query supporting the full capabilities of the ENTERPRISE query editor that are relevant to a single patient query.

### Assignment of variables and values

The editor should support assignment of variables and value to a query result set, the value being assigned directly by the user (e.g. assign value of ‘1’) or assigned from a value from the result set including:

1. Numeric value of a numeric record entry returned by the query
2. Average value derived from a number of record entries
3. Relative time period (to today) derived from the effective date of a record entry

### Creation of the formula function

The formula function consists of a code editor that provides:

1. A syntax checker, covering sufficient syntax requirements to code for complex formulas
2. A content checker making sure that variables are properly used and that return parameters are properly populated

The formula function should have at least three return parameters

1. A numeric value or a Boolean value
2. A list of display messages for use by the user in the consultation application
3. Entry Text to assign to the entry in the record

It should be noted that the text message can be created by the code within the code editor. For example:

Consultation Message (1)= “Score based on a systolic blood pressure of “\_bp

Where “bp” is a variable in the function.

### Selection of clinical code

The author must be able to assign a clinical code to the state. The system should associate the output of the state definition with that code.

### Editor Output

As well as storing the state definition within the state editor’s stored library, the use should be able to output the files containing the definitions as a file independently of any application.

### State definition library

A data store hosts a state editor library. The state editor library should support two types of access interfaces:

1. Update, edit and delete of the state definitions folders and workflow items
2. An access API to get the state definition query and function for use within the state calculator at query design time or run time
3. An access API to get the state definition for display within the Query editor that is part of the CEG enhanced query project.

## State calculator

The state evaluation component is a run-time software component that runs on a patient’s care record and generates state. It can operate within the data service on a population or within the data service on a particular patient, or in the consultation application on the local care record.

The calculator generates the state entry as determined by the state definitions and entity consists of:

1. Patient identifier
2. The code (or null)
3. Effective date and time
4. Availability data and time
5. Author of the state i.e. the system identifier that is creating this state
6. The value (or null)
7. Text messages (0 or many)

On calculation, the function will generate a coded entity, null, or a text only entity.

The following table illustrates the entity and entry creation when the function runs in batch mode or consultation mode.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Result | Entity | Batch Entry | Consultation mode |
| Boolean | 1 | Code+ entry text | Entry created | Entry created |
| Boolean | 0 | Null | No entry | No entry  + display message |
| Numeric | Not null | Code+ numeric value+ entry text | Entry created | Entry created  +display message |
| Numeric | Null | Null | No entry | No entry  +  Display message |

The calculator then posts the state content to the messenger for forwarding to the provider organisations.

## State messenger

The state messenger is a service that operates as a publisher to the RESOLUTION communication service.

It is responsible for

1. Generating the message in a standardised (FHIR) format
2. Assigning the relevant business process to the message (see below)
3. Posting the message to RESOLUTION or other standardised message handling service
4. Returning an errors to the ENTERPRISE query

Note. The RESOLUTION service uses the data sharing protocol to determine what then happens to the message. For example, the following may apply:

The publishing data controller is likely to be CEG.

The data sharing protocol may dictate that for a business process of “Risk score distribution” that:

1. Re-identification service will be used
2. Data is sent to the patient’s registered GP practice only
3. Data is sent both to the patient’s registered GP practice and is also stored in the EHR service (in CSU)

## Re-identification Service

In order to post results back to the patient’s practice it is necessary to find out who the patient is. This means finding the NHS number.

In addition, it is necessary to find the registered practice. This means finding the practice ODS code.

The CSU provides the forward hash mechanism to generate the pseudonymised identifiers for the patient and the practice.

RESOLUTION will call the re-identification service to obtain the identifiers. The identifiers will replace the pseudonymised identifier.

## State calculator Service

There is a need to calculate states in real time during the course of a consultation when the user is using a clinical system. The user is often using a data entry template or has entered some data into the record.

The clinical system may not support the formula, or else the formula needs to query data from other sources concurrently. In these circumstances, the system can call on a calculator service to return the state values in real time.

A service is required that accepts an inbound request for a specified state, that request containing:

1. State identifier (e.g. the return code)
2. Patient identifier
3. Current consultation coded data

The calculator service will run the calculation using the calculator and return the information via a message generated by the messenger service.