



OpenEyes - Care Pathways

Editors: G W Aylward

Version: 1.0:

Date issued: 10 September 2010



Target Audience

General Interest	✓
Healthcare managers	✓
Ophthalmologists	✓
Developers	✓

Amendment Record

Issue	Description	Author	Date
1.0	First issue	G W Aylward	10 Sep 2010



Table of Contents

Introduction	4
User experience	4
Considerations	4
Table structure	5
Care Pathways table:	5
Example Care Pathway	5
Integration	6
References	6



Introduction

OpenEyes provides a mechanism for implementing simple care pathways, which allows implementation of management guidelines, automation of clinical functions, and mandatory collection of outcome data. Care pathways are linked to an episode of care and a diagnosis. The data structure allows maximum flexibility in the design and functionality of any particular care pathway.

User experience

Care pathways are defined for each specialty, and are selected at the end of the first clinic visit, or when a diagnosis is made. While there will always be some patients who do not fit into one, it is envisaged that the majority of patient episodes will be associated with a care pathway. The user selects a pathway from a drop down menu, usually placed at the foot of the examination page.

The screenshot shows a software interface with two main sections. The top section, titled 'Conclusion:', contains four dropdown menus labeled 'Conclusion', 'Treatment', 'Outcome', and 'Timing'. Below these is a large, empty rectangular text area. The bottom section, titled 'Care Pathway:', contains three dropdown menus labeled 'Principal Eye', 'Diagnosis', and 'Care pathway'. At the very bottom of the dialog are two buttons: 'Save' and 'Cancel'.

If a drawing has been entered, and the drawing is linked to a diagnosis, then the appropriate pathway is selected automatically. Once selected, the pathway exists in the background, assisting with the construction of letters, selection of prescriptions, and collection of outcomes at the end of the treatment episode.

Considerations

The following list summarises the various characteristics required of care pathways, and their relation to episodes of care, and diagnoses.

- Every episode should, where possible, have a principal diagnosis associated with it.
- Each episode may have a care pathway associated with it.
- One care pathway may be associated with more than one diagnosis (the same as the episode diagnosis).
- One diagnosis may have more than one possible care pathway.



Table structure

The field definitions for the carepathways table are indicated in the following table

Care Pathways table:

Field	Type	Comments
carepathway_id	SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT	Primary Key, 65k
specialty_id	SMALLINT UNSIGNED	Foreign key referencing specialty
name	VARCHAR(64)	Name of care pathway, appears in selection menu
disorder_id	BIGINT UNSIGNED NOT NULL	Foreign key referencing disorders
description	VARCHAR(255)	Short description of care pathway
macro	VARCHAR(64)	Macro to run when writing letter
drugset	VARCHAR(64)	Drugset to run when writing prescription
conclusion	VARCHAR(120)	Text to go in conclusion text box
outcome	VARCHAR(64)	Rootname of outcome script

Example Care Pathway

The following table illustrates a care pathway for the management of traumatic hyphaema. The pathway is selected automatically when an anterior segment drawing of a hyphaema is entered, and the report button is pressed. Alternatively it can be selected manually.

Field	Value
specialty_id	11 (id of Paediatric Ophthalmology)
name	Hyphaema
disorder_id	75229002 (SNOMED code for traumatic hyphaema)
description	Recommended management for traumatic hyphaema
macro	hyphaema (letter macro for hyphaema)
drugset	milduveitis (drugset for mild anterior uveitis)
conclusion	Rest at home, topical steroids, needs gonioscopy and fundoscopy next visit
outcome	hyphaema (root or outcome collection script runs on discharge)



Integration

Once a care pathway has been selected and is associated with an episode, it can be interrogated by other OpenEyes modules to modify their behaviour. For example, the following pseudocode is used by the letter-writer module (OELetteroutEventClass) in order to automatically run a letter macro when launched.

```
if (session variable 'macro' is set) {  
    if (a macro of this value exists) {  
        Run the macro  
    }  
}
```

The outcome field contains the name of an additional script which is searched for and run automatically when a user closes an episode. This provides a mechanism for mandatory outcome data collection for specific episodes.

References

1. OpenEyes Snomed