X-Ray for D3 Cookie Y Emitters

Created by Nathan Caldwell, last modified by Oleksii Fedorenko on Aug 05, 2021

Use Cases for the database:

Track voiding and failure rate trends across dates, panel location, solder joint location.

Lookup data (including a link to the image) on an emitter in question at a future time

Data available from X-Ray Machine:

Timestamp

Panel Serial Number

Board/Pattern/Step number (Need to map to Panel Location letter)

Overall and largest void percentage and number of voids per each of 21 solder joints per emitter

Example of current output

https://lutrono365.sharepoint.com/:f:/s/D3transfertoCNC/Ep9t6DbKtEVMmHxG2ifLUoMBL5vv2gCiQ67xB8ZJkMROqA?e=jr5GqM

Data available from barcode scanning (should this be a PCAL tool?):

Panel Serial

Emitter serials belonging to that panel–but NOT location on panel

Data available from an emitter in question:

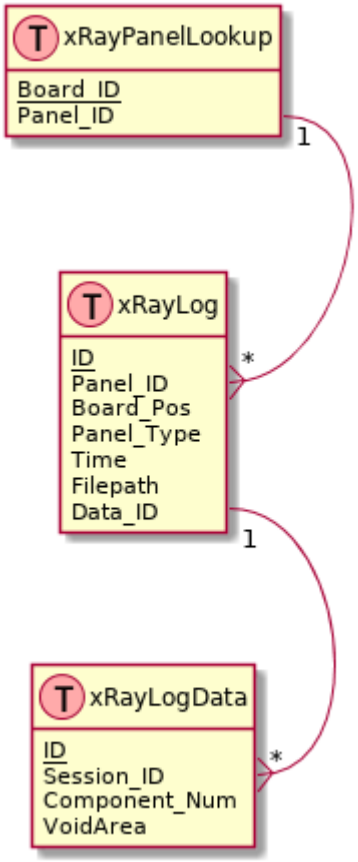
Emitter serial–but NOT panel serial

Panel location

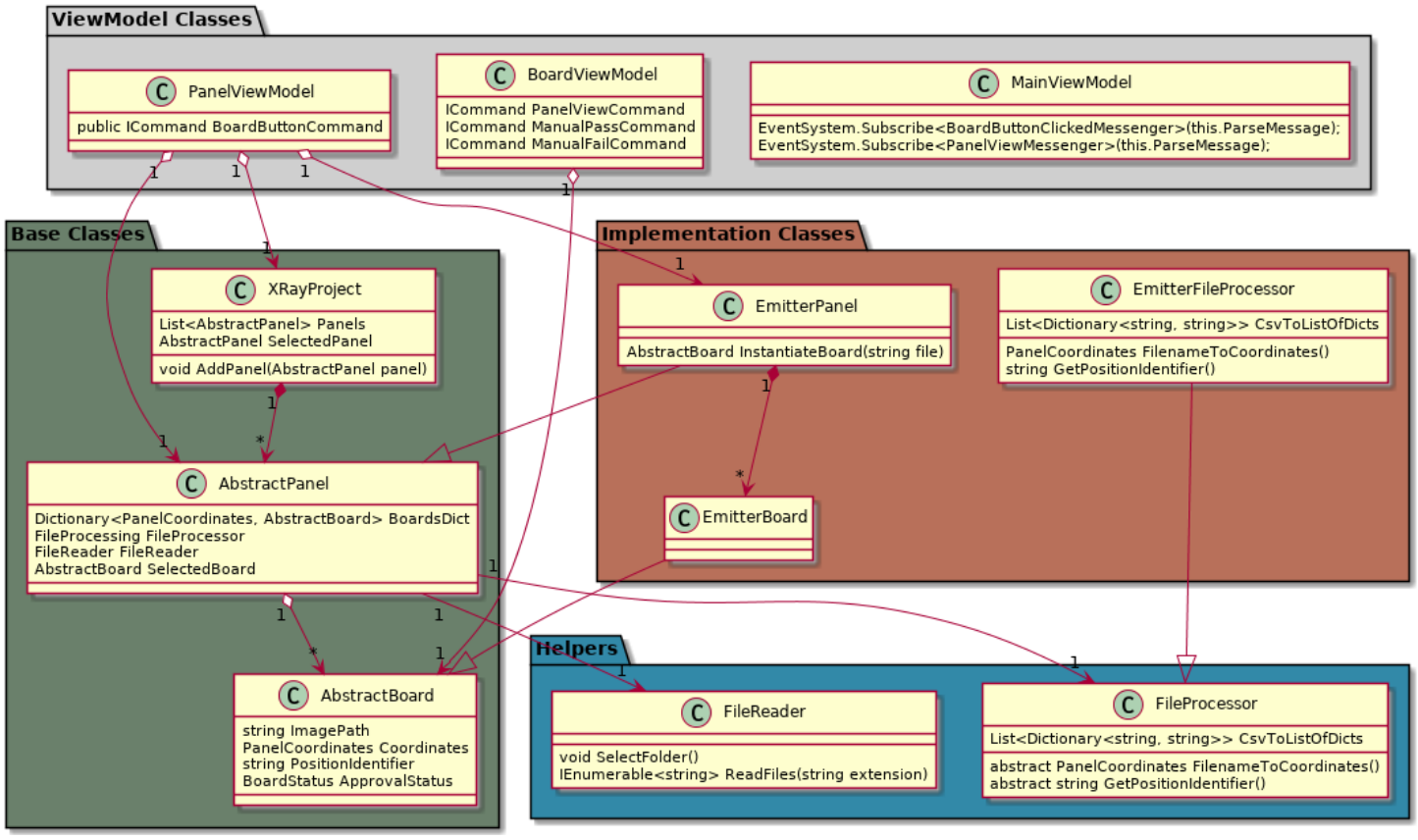
Requirements

I believe the python operator tool and the script may be combined. The GUI may contain the button that will run the script.

	Specification	Rationale	Notes
	Customer Requirements		
	The script MUST read the folder with all the CSV files and input that data into the database.		We are interested only in Overall Area Void (%) column of data in the CSV for each emitter.
	The database table MUST contain the data about the panel, emitter in rows, and data about the Overall Area Void (%) for each of 21 solder joints, and timestamp in the columns.		The same formatting as is used in the excel spreadsheet.
	The script MAY read the CSV with emitter serial numbers and add those to the DB.		
	The database MAY contain the panel and emitter serial numbers.		This can be implemented through a single table or through a relation of two tables.
	The GUI SHOULD have the button to execute the uploading of the data to the DB script.		
	The GUI MUST spacially display buttons representing emitters and highlight them if they need to be manually checked or thrown out.	Used for the manual checking	
	The GUI SHOULD display pictures of emitters on each of the buttons.	Used for the manual checking	
	The GUI SHOULD make buttons clickable and send the user to the screen with the enlarged emitter picture.	Used for the manual checking	
	The GUI MAY be automatically triggered.		
	System Functional Requirements		
	The scripted database writes MUST be compatible with local database server software (MS SQL 2014) and X-Ray software.	Have witnessed the X-Ray software lockup anytime a connection to a database is being attempted.	
	Performance Requirements		



Basic DB structure:



In addition to the database, @Ian Adams has a python operator tool for real-time pass/fail/manual inspection. Should this be 1 tool or separate tools?

No labels

1 Comment

Ian Adams

Yes I think it would make sense if the same tool accomplished both operations. Since we will be having the operator run the program for every set of data that the x-ray machine produces, we could have it set up so that without any extra steps, any time they run the program it would automatically send the data into the database behind the scenes while it is producing the visual for the operator. That way no data would miss getting uploaded to the database assuming the operator runs the tool each time like they are supposed to.