Poka Yoke Software Design : A Case Study on "Software for Vicarious Calibration of Optical Earth Observation Sensors"

** Yogdeep Desai*
*** Neelam Panchal*

Abstract

The total quality management (TQM) philosophy encompasses every facet of an organization to ensure "do it right the first time". Poka yoke or "mistake proofing" is a concept which helps in achieving "right first time, every time". The poka yoke has been successfully implemented in various hardware tools and components used in our daily life. Software, like any hardware tool, has also become a part of day-to- day activities like ecommerce and travel, just to name a few. Designing of any tool, hardware or software, developed by one for employing by the other, requires understanding of the user's perspective to ensure that the object is put only to intended use. Such interdependence in using a software tool mandates the implementation of poka yoke design in software development. Vicarious calibration procedure for Earth observation remote sensing system involves data from various sources and in various formats, which needs to be compiled and pre-processed before the analysis and generation of a calibration coefficient takes place. The vicarious calibration process also uses software utility for ensuring consistency and timeliness in delivery of results. This paper explores various elements of vicarious calibration software, which can be developed using poka yoke technique to ensure that such diverse data are pre-processed independently by members of the calibration team without making any unintended mistakes arising out of unintelligible software design implementation.

Keywords: **vicarious calibration, Poka yoke, radiance**

JEL Classification: C88, C93, L15, L86, M11, M15, M19

Paper Submission Date : July 12, 2014 ; Paper sent back for Revision : September 26 , 2014; Paper Acceptance Date : October 16, 2014

The beauty of a software design lies in its simplicity, fluidness, intuitiveness, and transparency to the user of the software (A Techie's Notebook, 2012). The user is expected to be guided intuitively by the virtue of software design, keeping the user free to concentrate on the problem-at-hand rather than toiling with the complexity of the software. A software is meant to be used repetitively over a long period of time, and hence, the one-time effort in the design of the software goes a long way in easing the task for which it is intended to be designed. The vicarious calibration of optical remote sensing sensors involves collection and interconnection of inputs coming from various sources, in various formats, generated or acquired at different times. The inputs are partially measured by the analyst/user and are partially acquired from other sources.

Poka yoke- a mistake proof- design pertains to Patil , Parit, and Burali (2013), preventing errors or mistakes from occurring at the process level. It is also considered an important tool from the total quality management (TQM) perspective (Kanji & Asher, 1996) . Thus, a software with poka yoke philosophy incorporated in the design not only relieves the analyst from making inadvertent mistakes, but can also help in the smooth flow of the process, resulting in consistent and timely generation of results.

* *Scientist (SE)*, Space Applications Centre (SAC-ISRO), Indian Space Research Organization, Ahmedabad. E-mail : yogdeep1863@yahoo.co.in
** *Faculty,* Institute of Management, Nirma University, Ahmedabad. E-mail : nilamcpanchal@gmail.com

Figure 1. Three Types of Poka yoke (a) Control (b) Warning (c) Shutdown

Source: http://www.spyghana.com/wp-content/uploads/2012/08/sim-card.jpg
http://thumbs.dreamstime.com/z/fasten-seatbelts-277833.jpg
http://www.pawlikautomotive.com/wp-content/uploads/2011/05/Blown-fuse.png
http://www.wheelchairdriver.com/fuse.htm/fuse.h4

What is Poka - Yoke ?

This technique is mostly used in manufacturing industries (Nataraj, n.d.). Poka Yoke is a strategy of "mistake - proofing" to prevent inadvertent mistakes and prevent defects . The poka yoke device or a fail - safe procedure is developed on the principle that mistakes can occur, but should not occur again (Nataraj, n.d.). One of the important principles of TQM is "do it right the first time". Poka yoke in software design ensures that the input from users is "right the first time" (Miralles, Holt, Marin-Garcia, & Canos-Daros, 2011), which will increase the "productivity" without avoidable haste-which can only increase rework. The concept of poka yoke was developed by Shigeo Shingo in 1961 (Miralles et al., 2011), which resulted in stopping of a process as soon as the defect occurred, identifying the source of the defect, and then preventing it from happening again. A real-life example of poka-yoke design is shown in the Figure 1. The notched corner of a mobile phone's SIM card prevents the user from wrongly inserting it in the slot and being "right the first time".

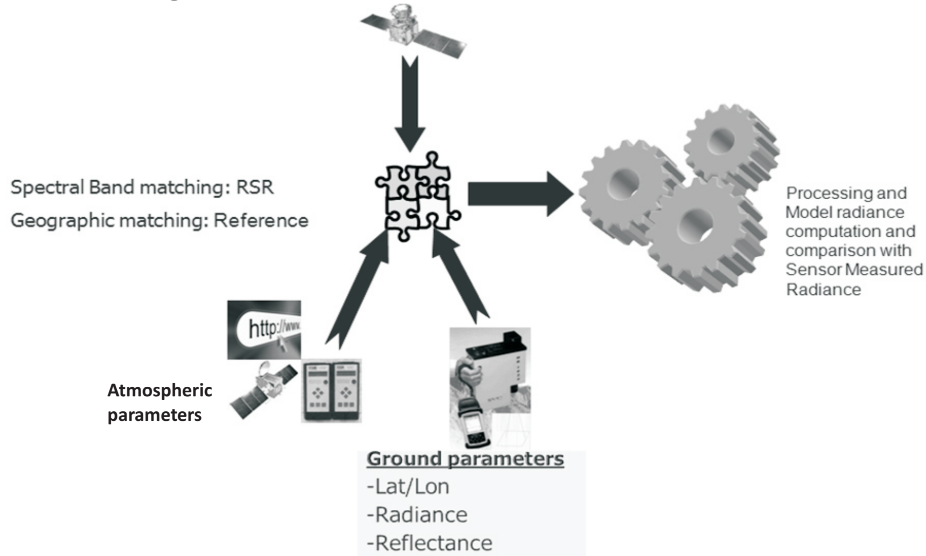
The mistake-proofing can take any one of the following three types (Miralles et al., 2011): Control poka yoke, warning poka yoke, and shutdown poka yoke. The design of a mobile SIM card is an example of control poka yoke, where the shape of the SIM card and the corresponding slot makes it impossible for the user to insert it in any other direction than what is right. Warning poka yoke is used where the user is made conscious of something "not right" and expects the user to take an appropriate action. The seat-belt sign in automobiles is an example of such warning poka-yoke, which alerts the driver that the seat-belt has not been put on, but allows the process of driving to continue even if the driver has not taken any action on the warning. The shutdown method works on the principle of "prevention is better than cure," and shuts down the process as soon as the process control parameter goes out of acceptable control limits, thus preventing a bigger damage. The electric fuse blows off as soon as the voltage crosses a pre-defined limit; thus, preventing damage to the electrical equipment using the electrical supply.

Poka yoke is also useful in activities which are monotonous (Miralles et al., 2011) for the operator and tends to make the operator complacent over a period of time, creating an opportunity of neglect on the part of the operator. The automatic-water-tank-leveller is a system which automatically ensures that the overhead water tank of a building does not overflow as well as does not remain empty due to the negligence of the pump operator. Through shutdown of poka yoke, it is also designed to ensure that the pump does not dry-run.

Vicarious Calibration Method

The "vicarious" or secondary calibration is a post-launch calibration process (Chen, Zhang, Zhang, & Zhang, 2014) for Earth Observation (EO) sensors to ensure consistent results throughout the life time of the EO mission.

Figure 2. Vicarious Calibration Process in a Nutshell



'Calibration' is defined as an operation, that under specified conditions, in a first step, establishes a relation between the quantity values with measurement uncertainties provided by measurement standards and corresponding indications with associated measurement uncertainties and, in a second step, uses this information to establish a relation for obtaining a measurement result from an indication (International Vocabulary of Metrology, 2006).

An Earth Observation system undergoes laboratory calibration prior to launch to establish a relationship between the quantity values with measurement uncertainties provided by measurement standards - which is a calibration source, and corresponding indications with associated measurement uncertainties-the output from Earth's observation systems detectors in the form of digital counts. This relationship is stored in the form of calibration coefficients for obtaining a measurement result from an indication. Over a period of time, and even during the launch, due to the harsh outer-space conditions and normal wear and tear of the system, the laboratory derived coefficients may not hold good (Teillet , Fedosejevs, Gauthier, O'Neill , Thome , Biggar, Ripley, & Meygret, 2001). This alteration in relationship between the input energy and system response to this incident energy needs to be constantly corrected for it to derive a consistent result from the EO system throughout its designed mission life.

The reflectance-based vicarious calibration procedure (Gilead & Karnieli, 2004) involves extensive ground truth measurements of the surface spectral reflectance of ground and measurement of atmospheric parameters, synchronous to satellite overflight using calibrated ground-measuring as well as atmospheric-parameter-measuring instruments. Using the radiative transfer code (Chandrasekhar, 1960) to model the interaction of radiation with atmosphere, the ground measured reflectance is converted to modelled top-of-atmosphere (TOA) radiance. Coefficients are derived as a relationship between the modelled TOA radiance and the top-of-atmosphere (TOA) target radiance measured by the sensor. These computed coefficients can be used in place of or in addition to those derived during the pre-flight laboratory calibration exercise (Desai, Srivastava, Bhavsar, & Kartikeyan,2013). The vicarious calibration process is depicted, in a nutshell, in Figure 2.

The ground measuring instruments consist of hyper spectral (Spectra Vista Corporation, 2010) radiance/reflectance measurement capability. These measurements are required to be integrated for the respective spectral bands of the sensor being calibrated using the sensor's spectral response function (also called relative spectral response-RSR) to get band integrated radiance (BIR), which is the input for deriving TOA reflectance required by RTC to convert to TOA radiance.

Poka Yoke Design of Vicarious Calibration Software

The dimensions of software quality (Kan, Basili, & Shapiro, 1994) are defined in terms of software standards like IEEE 12207. The poka yoke aspect of the software is the operator's perspective and goes a long way in assuring that the software delivers up to the level of quality defined by software standards.

Smith and Mosier (1986) emphasized the need for data entry validation. The validation is a form of control poka yoke, where the designer of the software checks for the correctness of entry before proceeding to the next step. The present design of vicarious calibration software, which is executed by analysts/users is based on certain principles and assumptions:

(1) For the purpose of clarity in addressing the information in this paper, the vicarious calibration of a given sensor using the corresponding ground measurement campaign will be termed as a Project, that is, a Project consists of processing and analysis of ground measured data, atmospheric parameter data, as well as sensor data products for which calibration is being performed; the Project has to accomplish many Steps before completion.

(2) A Session of cal/val program begins on invocation of the program and ends with Exiting the program. Thus, a Project can be completed in multiple Sessions.

(3) A Session is composed of multiple Steps related to the Project, and the Session can only end after successful execution of a Step.

(4) All Steps for a Project should be taken only through the cal/val software - called the Online mode; the cal/val software is a Closed-loop system where no operation is done outside the software related to a given Project of vicarious calibration chain (e.g prior copying of data related to the project, in particular, area of disc should not be anticipated while designing the software).

(5) Initially, the user must be given the freedom to choose the media for the data brought from an external source like ground measuring instrument or data product on a compact disc. But once the data is extracted/downloaded from the chosen media using the software, the software should track the path of data downloaded on the system, and the user need not look for the data directory for further processing.

(6) The access to the data downloaded for subsequent processing should be through information already available with the user, which is external to the software design and relates only to Project; the user need not remember the path of the data to access it.

(7) Once a value for an input is given for a particular Step, the subsequent steps for a given session should be replaced by drop down menus, wherever possible, for users to select rather than enter again.

(8) For data entry to a particular Step, display the most probable default values for increasing the speed without the user having to remember the same.

(9) The validation related to a range of entered values, format, and so forth must be carried out after each entry.

(10) At the successful completion of every Step, the action taken (e.g an output file generated) must be shown to the user along with the necessary information (e.g the path to which the file is saved).

(11) The design should facilitate the completion of Project in multiple Sessions through maintenance of history of the Steps already accomplished until the previous Session of a given project. With these assumptions and principles, the software for vicarious calibration is designed. The underlying aspect of the vicarious calibration process as derived from the Figure 2, and can be broken into the steps given in the Table 1. These steps then become the input for the designer of the software.

Table 1. Vicarious Calibration Process and Software Design Aspects

Step number	Underlying process	Operational Aspect to be addressed in software design
1	Extract ground measured data in the form of hyper spectral target radiance and reflectance values and corresponding latitude and longitude of the place of measurement called the training site.	How does the analyst access the ground measured date? Where does the extracted data go? How does the analyst name the output directory?
2	Convolve these hyper spectral radiance data with sensor's spectral response function to get Band Integrated Radiance (BIR).	How does the user recall the directory generated in the previous step? The RSR of which sensor should be used?
3	The data from satellite over region of interest (ROI) for same date and time are extracted and kept separately along with sun and satellite angles. The process of mapping latitude and longitude of site with the scanline and pixel from the image involves image-to-image registration with respect to a geo-referenced chip template of the same region. The sensor measured radiance (SMR) is then extracted for those pixels where ground measurements were taken.	How does the user recall the directory generated in the previous step? How does the analyst access the data product? Where does the downloaded satellite image and meta information get stored?
4	The process of converting ground measured radiance to top-of-the-atmosphere (TOA) radiance involves the radiative transfer code (RTC). The RTC takes input from another device- the sunphotometer / ozonometer- which have its own wavelength filters and the BIR generated in step 2 to compute TOA radiance.	How does the user recall the directory generated in the previous step? How does the analyst access data of atmospheric parameter corresponding to date and place as that of satellite data? Where does modelled TOA radiance get stored?
5	The radiance measured by sensor (SMR) and TOA radiance generated through RTC are used to derive new set of calibration coefficients.	How does the user recall the directory generated in the previous step?

The vicarious calibration process can be responded through a variety of software design. The software design inspired by poka yoke philosophy not only imparts flexibility to the analyst in terms of running of the software at convenience, but also gives a handle to getting back to track without much of an effort. It also ensures that the software can be run by any member of the team seamlessly without having to consult the other team members on status of various stages. The design needs to take the following basic assumptions into consideration:

- Each member is aware of the sensor being calibrated and its date of pass,
- The name of site and date when measurements were taken.

At the outset, the poka yoke design needs to understand the user's requirement or expectations from the software (Table 2). Once the user's requirement is known, identify steps where the scenario can fail. Design the software to avoid such a failure. It need not be a complex algorithm; just a simple check or warning can alert the user of impending error. The poka yoke design requires thorough understanding of the user's setting, and hence, it is worthwhile to enlist the poka yoke design step for future use after ensuring its effectiveness in preventing mistakes by users under similar situations. The Table 2 gives a comparison between a non-poka yoke design and a poka yoke design of vicarious calibration software. The Figures 3- 9 are snapshots of vicarious calibration utility with poka yoke design mentioned in the Table 2. Each option in the main menu is a Step to be performed for completion of vicarious calibration Project of a particular sensor:

(1) Ground Data Extraction: Draws out information like ground radiance, latitude/longitude of measured point, and so forth concerning the vicarious calibration process from the instrument measured data file and stores in required format for further processing.

Table 2. Comparison Showing the Poka Yoke and Non-Poka Yoke Method of Satisfying User Requirement

User Requirement	Non- poka yoke	Poka yoke	Type
Accessing ground measured data for extraction-A process of extracting information from vendor format to user required format.	Create a directory with name given by programmer-and hardcoded in software- in the predefined area of the disc. Copy all datasets in the directory so that the software reads the data from hard-coded directory name.	Allows the analyst the freedom to browse the media in which the analyst holds the data; this saves the analyst from remembering the name of the directory to be created and the step of copying the files before extraction begins.	
Creating output directory for storing extracted information.	Allows the analyst to key-in the name of satellite/sensor/site and select date from system calendar function; this can create erroneous and multiple strings for names arising out of misspelt words.	Allows the user to select the names of satellite, sensor, site from a drop down menu-created from a configurable file. This Control Poka yoke keeps consistency in names. At this stage, the date can be selected from a system's calendar function; the output directory name is constructed by the software itself based on these inputs. This process is transparent to the user.	Control poka yoke
Accessing/recalling the directory subsequently for processing at a later date or by other team member.	Displays the list of directories created till-date for the analyst to select; over a period of time, this list can be very long, making the selection of directory cumbersome and the analyst can make an inadvertent mistake in selecting the same which results in generating untraceable erroneous results.	The analyst is allowed to select the name of satellite, sensor, and site name. Even the date is displayed only from that available from existing directories rather than from system calendar function. This ensures that the analyst does not have to hunt for existing possible combinations and is guided through display of controlled options; this also reduces load on validation.	Control Poka yoke
Status of each step	No information about the progress of each stage; each stage runs independently; it is the responsibility of the user to be judicious in selecting the folders.	The message of completion is displayed after every stage; the creation of folder follows Just-in-Time principle and hence, existence of a folder signals the completion of a particular process.	

(2) BIRC : Generates band integrated radiance data using the data extracted in Step 1 and the RSR of the sensor.

(3) Product Processing: The sensor measured data in the form of data product is downloaded in the disc from selected media. The meta information and binary image files are sorted separately. This Step also displays the image along with corresponding geo-referenced template for image to image registration for geo-referencing of the input image. Coefficients relating the scanline / pixel of the image and latitude/longitude of the region are generated at this stage.

(4) Scan Pixel from Coefficient: Using the coefficients generated in preceding Step 3 and latitude/longitude of ground measured data stored in Step 1, all ground measured points are assigned scanline and pixel number of the image.

(5) Integrated Output : This Step aggregates the information collected so far to generate an integrated radiance value from ground measured for scan line and pixel of the image assigned in Step 4.

Figure 3. Main Menu Showing Various Stages of Vicarious Calibration Process

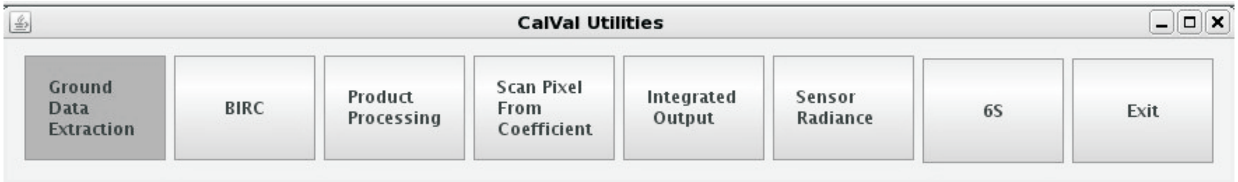


Figure 4. Browsing Facility for Media Selection and Information for Output Directory

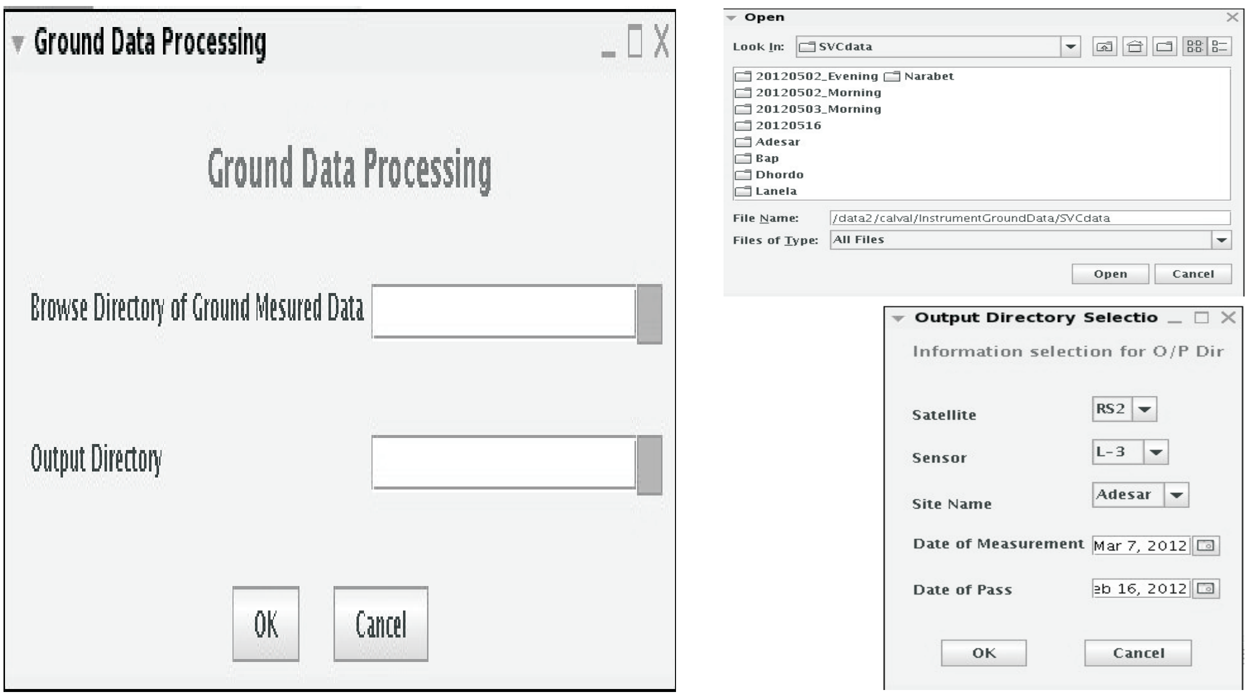


Figure 5. The Date Selection Through Drop Down Menu

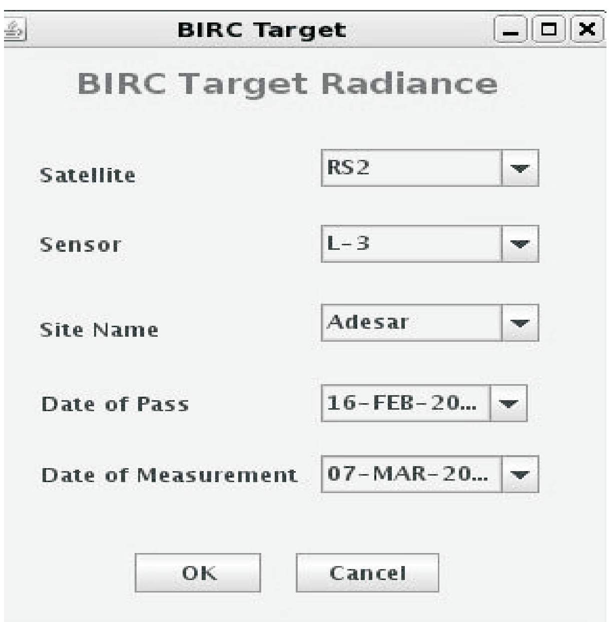


Figure 6. Case Showing Control Poka Yoke

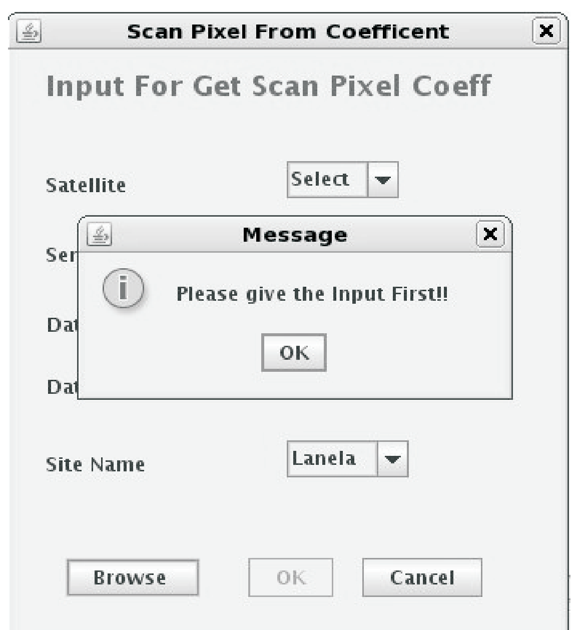
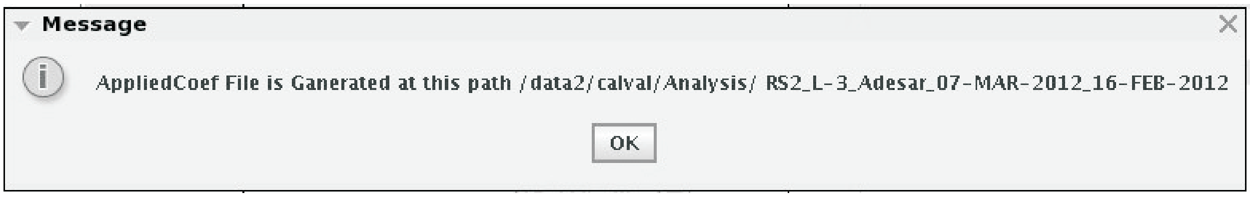


Figure 7. Completion Message with Output Directory Name



(6) Sensor Radiance : The sensor measured radiance (SMR) for the same scan line/ pixel values as that in Step 5 is computed here.

(7) 6S: This step uses atmospheric parameters measured by Sunphotometer/Ozonometer and converts the ground measured reflectance to modelled Top-of-Atmosphere radiance (MTR). It compares the SMR and MTR to compute calibration coefficients relating the two.

Figure 8. Allows Browsing for New Product

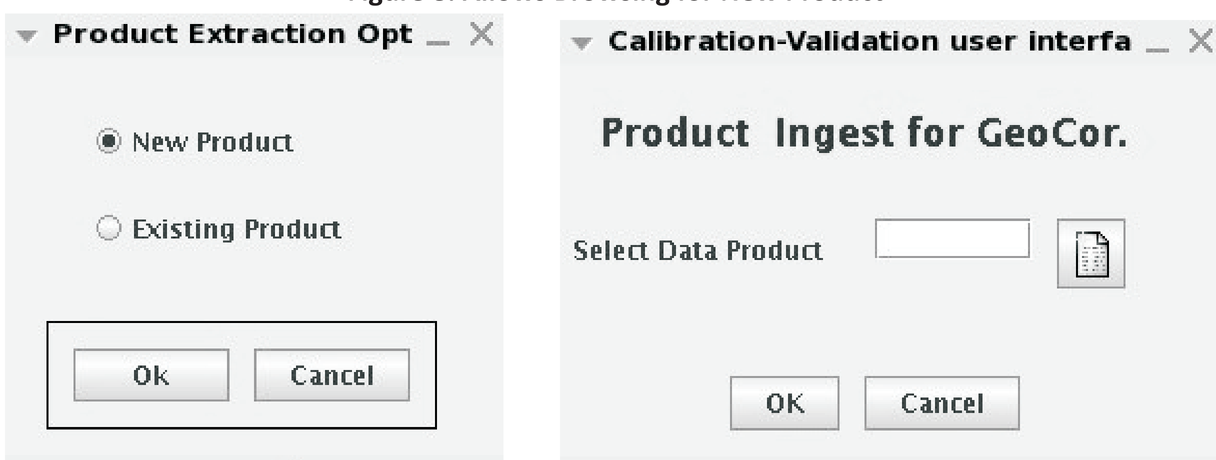
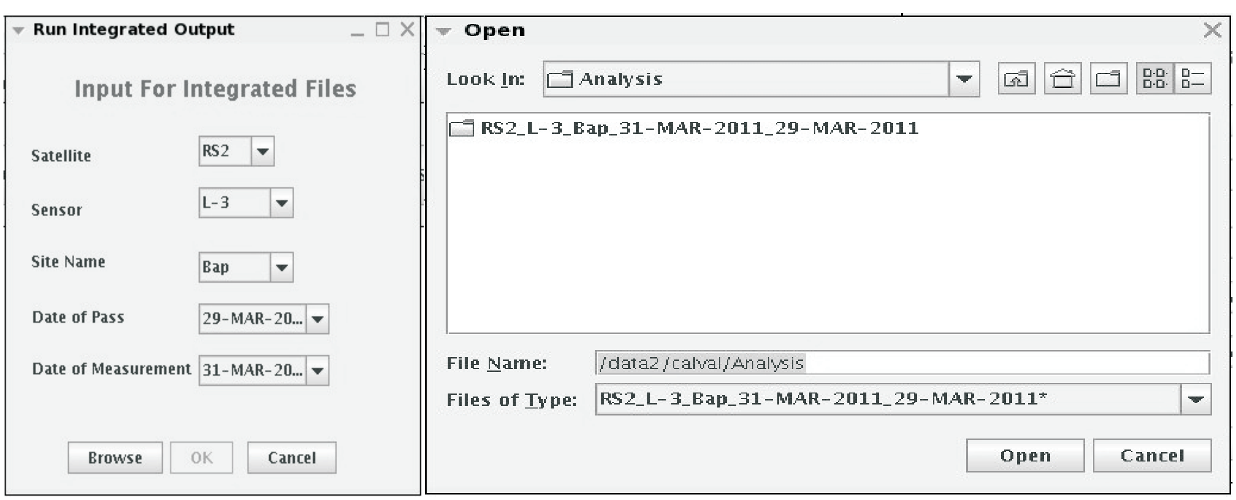


Figure 9. Control Poka Yoke: Guides to the Directory Based on Selected Inputs



Conclusion and Implications

The principle of poka yoke is about making a process mistake-proof. The mistake-proofing can be achieved by one or a combination of three poka yoke types: warning poka yoke alerts the user before something wrong is being done, but allows the process to advance at the discretion of the user. The shutdown poka yoke works on the principle of "prevention is better than cure" and thus, does not allow the process to proceed and simply terminates the process, thus preventing greater damage. The control poka yoke guides and controls the user's action only in the acceptable direction and restricts any other non-acceptable action. The vicarious calibration of Earth observation sensor involves compilation, matching, and processing of inputs from various sources and formats. These similar looking inputs require a poka yoke design of vicarious calibration process, which can assist the analyst through the various stages. The poka yoke principle of designing the software results in avoidance of unintended error on the user's part and ensures consistent reliable results for the analyst.

This paper brings out the importance of poka-yoke design of software, which will help the developer in giving greater satisfaction to the user of the software by preventing inadvertent mistakes which can result in disastrous outcomes in spite of having the best possible algorithm in place.

Limitations of the Study and Scope for Further Research

This research was carried out only on scientific data processing. Users' requirement of a commercial software can be taken up as a separate study. The software quality from supplier-input-process-output-customer (SIPOC) perspective needs to be investigated, especially when the software is designed and developed for internal users.

References

A Techie's Notebook (2012, July 22). *POKA YOKE - Applying mistake proofing to software* [web log post]. Retrieved from http://techie-notebook.blogspot.in/2012/07/poka-yoke-applying-mistake-proofing-to.html

Chandrasekhar S. (1960). *Radiative transfer*. New York: Dover Publication.

Chen, Z., Zhang, B., Zhang, H., & Zhang, W. (2014). Vicarious calibration of Beijing-1 multispectral imagers. *Remote Sensing, 6*(2), 1432-1450. doi:10.3390/rs6021432

Desai, Y., Srivastava, S. S., Bhavsar, V. R., & Kartikeyan, B. (2013). *Vicarious calibration for LISS-3 (Resourcesat-2), part II: anomalies & their removal*. SAC/SIPA/SPDCG/IAQD/CALVAL/TN-89/Sep'13.

Gilead, U., & Karnieli A (2004). *Locating potential vicarious calibration sites for high-spectral resolution sensors in the Negev Desert by GIS analysis*. London, UK : Taylor & Francis Group.

International Vocabulary of Metrology (VIM). (2006). *Basic and general concepts and associated terms* (VIM 3rd edition). Final draft 2006-08-01, JCGM/WG 2 Document N318.

Kan, S.H., Basili, V.R., & Shapiro L.N. (1994). Software quality: An overview from the perspective of total quality management. *IBM Systems Journal, 33*(1), 4-19.

Kanji, G. K., & Asher, M. (1996). *100 methods for total quality management*. New Delhi : SAGE Publications.

Miralles, C., Holt, R., Marin-Garcia, J. A., & Canos-Daros, L. (2011). Universal design of workplaces through the use of Poka-Yokes: Case study and implications. *Journal of Industrial Engineering and Management, 4* (3), 436-452.

Nataraj, K. (n.d.). *How to use Poka-Yoke (mistake proofing) technique to improve software quality*. Retrieved from http://www.softwaretestinghelp.com/poka-yoke/

Patil, P. S., Parit, S. P., & Burali, Y.N. (2013). Review paper on "Poka Yoke: The revolutionary idea in total productive management". *Research Inventy: International Journal of Engineering and Science, 2*(4), 19-24.

Smith, S. L. & Mosier, J. N. (1986). *Guidelines for designing user interface software*. Massachusetts, USA: The MITRE Corporation Bedford. ESD-TR-86-278.

Spectra Vista Corporation. (2010). *HR-1024/ HR-768 user manual revision* 1.10, New York 12603.

Teillet, P.M., Fedosejevs, G., Gauthier, R. P., O'Neill, N.T., Thome, K.J., Biggar, S.F., Ripley, H., & Meygret, A. (2001). A generalized approach to the vicarious calibration of multiple Earth observation sensors using hyperspectral data. *Remote Sensing of Environment, 77*(3), 304 - 327.

Untitled illustration of a blown fuse (n.d.). Retrieved from http://www.pawlikautomotive.com/wp-content/uploads/2011/05/Blown-fuse.png

Untitled illustration of a fuse (n.d.). Retrieved from http://www.wheelchairdriver.com/fuse.htm/fuse.h4

Untitled illustration of a seat belt. (n.d.). Retrieved from http://thumbs.dreamstime.com/z/fasten-seatbelts-277833.jpg

Untitled illustration of a SIM Card. (n.d.). Retrieved from http://www.spyghana.com/wp-content/uploads/2012/08/sim-card.jpg