31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

A Flexible System to Capture Sample Vials in a Storage Box – the Box Vial Scanner

Steven E. Nowakowski, Kenneth R. Kressin, Steven D. Deick

Abstract—**Tracking sample vials in a research environment is a critical task and doing so efficiently can have a large impact on productivity, especially in high volume laboratories. There are several challenges to automating the capture process, including the variety of containers used to store samples. We developed a fast and robust system to capture the location of sample vials being placed in storage that allows the laboratories the flexibility to use sample containers of varying dimensions. With a single scan, this device captures the box identifier, the vial identifier and the location of each vial within a freezer storage box. The sample vials are tracked through a barcode label affixed to the cap while the boxes are tracked by a barcode label on the side of the box. Scanning units are placed at the point of use and forward data to a sever application for processing the scanned data. Scanning units consist of an industrial barcode reader mounted in a fixture positioning the box for scanning and providing lighting during the scan. The server application transforms the scan data into a list of storage locations holding vial identifiers. The list is then transferred to the laboratory database. The box vial scanner captures the IDs and location information for an entire box of sample vials into the laboratory database in a single scan. The system accommodates a wide variety of vials sizes by inserting risers under the sample box and a variety of storage box layouts are supported via the processing algorithm on the server.**

I. Background

Tracking sample vials is a critical task in a research environment. To carry out an experiment the researcher must know what samples are available and be able to retrieve them. Efficient and robust sample tracking can have a large impact on research productivity. Moreover, some of the laboratories perform a high volume of initial processing on samples before they are sent off to individual research areas. Protocol requires samples to be tracked as they are transferred between labs. Often the sample vials are accumulated in the processing lab then transferred in batches to the research laboratories.

Researchers tend to have a strong preference for a particular vial and storage box for their samples. Standardizing the laboratories on a small set of containers is not an option. Similarly, switching all the laboratories to pre-labeled vials is not a feasible option. Vials stored in the laboratories encompass a large range of sizes holding up to 5ml of fluid with 1-3ml being most common. The vial bottoms are round, flat or cone shaped with caps that are flat, rounded, or even have a brim. Vial height varies from 2 to 3.75 inches in height.

Fig 1. Vials commonly used for sample storage

Vial storage boxes hold 25, 81 or 100 vials. The boxes are constructed of various plastics, or cardboard. In the box the vials stand on end separated by a grid of plastic or cardboard, or in the case of a 100 vial box just pegs. One lab stores tissue samples in disc shaped vials stood on edge in a 3 X 8 layout.

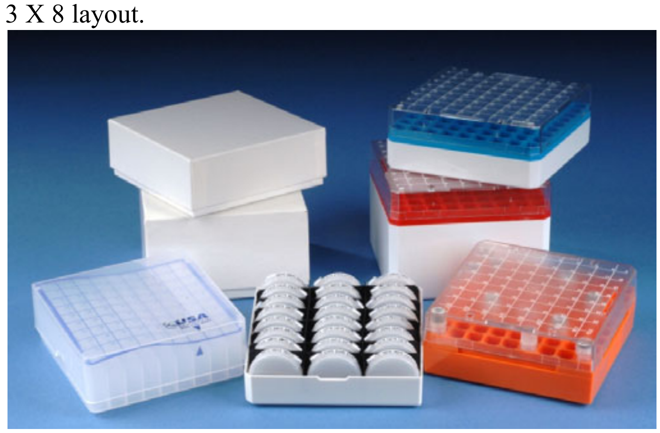

Fig 3. A selection of vial storage boxes

Vials are usually labeled on the side where a label can be larger and hold more readable data. In a storage box only the vial cap is visible, so the system requires another label on the cap. The vial cap is less than 0.375 inches in diameter. A linear barcode with the required data encoding would not fit

Manuscript received March 27, 2009.

S. E. Nowakowski is with the Division of Engineering, Mayo Clinic, Rochester, MN 55905, USA (nowakowski.steve@mayo.edu)

K. R. Kressin and S. D. Deick are with the Division of Engineering, Mayo Clinic, Rochester, MN 55905, USA.

in the limited space, so a 2 dimensional DataMatrix barcode was chosen.

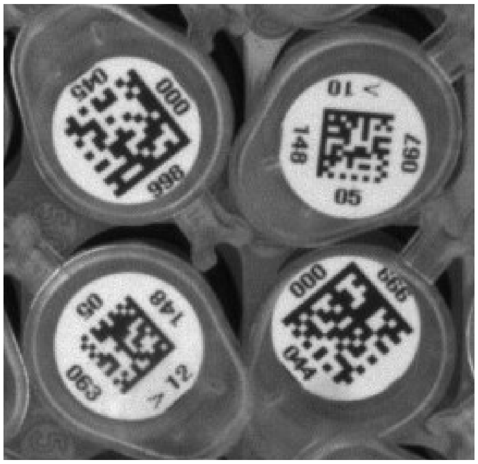

Fig 2. DataMatrix 2D barcode labels

The Box Vial Scanner (BVS) is a second generation system based on the Vial Archival and Tracking System (VATS) built by the Division of Engineering to track sample vials for two specific labs. VATS demonstrated the feasibility of tracking vials with barcodes placed on the vial caps and is in daily use in a research lab.

II. User Needs and Requirements

The project began when the director of a sample processing lab (a "core" lab that serves primary research labs), approached the Division of Engineering to see if the VATS system could be modified to interface with a new database. We started development by gathering use cases. The first use case was: "the user scans a box of sample vials then the box ID, vial IDs and location data are sent to the laboratory database". This use case is notable because it's the *only* use case (outside of maintenance). We thought maybe this use case is too high level and needs to be broken down. So, we broke it down for each type of vial and box configuration. However, these are not really use cases but test cases and were later used as such.

In contrast to use cases, the user *goal* is: "the user captures sample vial IDs and location data for the box of vials into the laboratory database". The user goal is much more powerful than a use case for developing a streamlined product. Goal based user needs analysis emphasizes simplifying the users' interaction with the system. The user doesn't need to be involved in intermediate steps such as selecting a box layout or viewing the scan output before sending it to the database. These are tasks the system should accomplish for the user if at all possible. The user should scan a box and view the output in the database.

Based on the user goal, the user interface became: insert a box to scan, push the scan button, and remove the box after the lights flash. Afterward, the user can manipulate the data in the database. Of course, the system must be robust so a user can trust their scan will be captured accurately.

Engaging users at the early design stages is incredibly valuable, but abstract discussions fail to reveal important needs and constraints. Bring mockups or props to the user meetings. An early mockup placed the box identifier label on a vial location, so it could be seen from above by the barcode scanner. The users did not want to lose a storage location and suggested we add a second barcode reader to read the label on the side of the box instead. We took on the challenge of reading the box label. A second barcode reader would have added significant complexity and cost to the system. Instead, we added a mirror so the barcode reader viewing the top of the storage box could also see the label on the side of the box. We gained a storage location in the box and the main box label was the same label being read by the scanner. This removes the possibility of a mismatch.

III. System Design

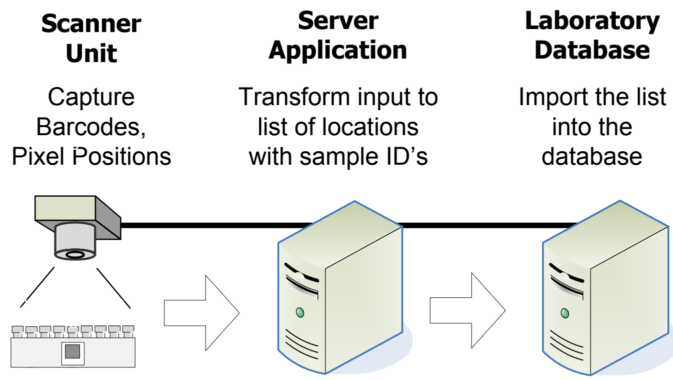

Fig 4. BVS system function diagram

Developers may tend to start with a PC based design to interface a sensor or subsystem. This may be a defacto for many. But if an Ethernet connection to the equipment is available, a server based design has many advantages:

- Highly productive software development environment
- Plethora of tools, libraries
- Support from IT for backups, etc
- Requirement to transfer captured data to network resources
- Flexible amounts of processing power, storage
- Single location for support, updates, tracking

The single location for support, updates can greatly reduce developer time spent on maintenance and on testing since only one environment needs to be tested. Much equipment has a network interface available.

Good object oriented software design has a function in its place and only one place. The same holds true for system architecture. A parallel rule is to use each component for the function it does best. Our system splits into two subsystems: the scanner units and the server application. The scanner unit captures the barcodes while the server application transforms the scanner data into a list of locations in a storage box with vial IDs.

IV. THE SCANNER UNIT

The function of the scanner unit is to capture each vial ID and its pixel position. The scanner unit is focused on this one task and must do it successfully all the time. The task includes:

- Supporting the barcode reader
- Holding the box of vials in position,
- Lighting the barcode labels,
- Triggering on the user press of the scan button,
- Reflecting the box label into view of the barcode reader

The scanner unit is focused on reading barcodes. Consequently, all the information on the scanner unit is represented by barcodes. The scanner unit serial number is a barcode visible to the barcode reader. The unit serial number is read in each scan and sent along with the other barcode data. The unit serial number allows the source of each scan to be tracked and the barcode position can be used to verify the unit calibration.

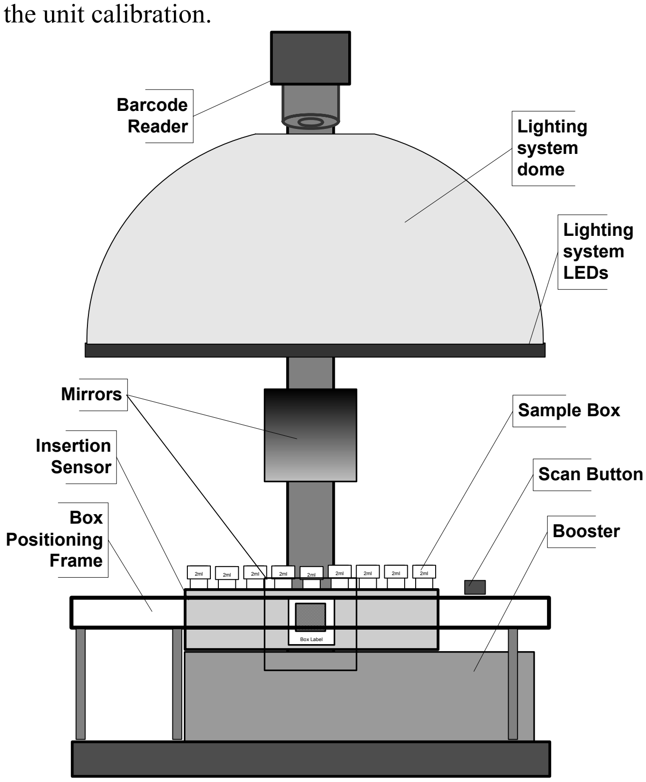

Fig 5. Scanner Unit Functional Parts

The barcode reader is made by Data Logic. This is one of the few barcode readers that can read 100 barcodes at a time. It's fast, reliable, highly configurable, and networked, with good input/output controls.

Robustly reading the barcodes under varying conditions requires a systematic approach: Light flows from source to the labels then the image flows to the barcode reader. Ensuring all 81 labels in a box read every time demands a highly consistent level of illumination on each label from the center of the box to each edge and corner. A low angle of illumination is required to shift the surface reflection away from the barcode reader. The lighting system consists of a ring of 21 white LEDs reflected into a plastic dome. Meanwhile, the box side label is lit with a separate lighting system. The box side label is reflected off two mirrors so it is not inverted when seen by the barcode reader. The box label light turns on to indicate the box is fully inserted. The main ring of lights fires when the scan is triggered.

The labels must be generated and printed in a size and format that encodes the vial ID, fits on a vial cap, can be printed with our existing printers, and is large enough for the sensor read robustly. We chose a 14 x 14 barcode with 14.8 mils module size. We developed a unique label for the disc shaped vials with redundant barcodes distributed around the diameter so the vial could be read from any angle.

The issue of variation in container height is treated with a very simple solution – a booster to place under shorter boxes. Active design solutions were explored, but the simplicity and intuitive user application of the booster made it the best design option.

V. THE SERVER APPLICATION

The server application implements the unique value of the system and provides the glue to make the parts into an integrated system. The server application design embraces a flow through model. Data flows through each functional block of the application being transformed along the way. Since each set of scan data is fully independent, no communication between the data sets is required. The transformation in a block may not always be the same but the input and output objects are the same.

Before the data can flow in the pipeline we need to get it from the scanner units. The scanner communications block receives the data from the scanners via TCP/IP. The scanners initiate the connection to the server application and send the barcode data in several chunks. After a full set of data is received, the data is wrapped in an object called Scan Data and passed to the first functional block to start the processing.

In the first block, the barcode data from the scanner is parsed into barcode IDs paired with the pixel location. The data is also validated for completeness. Note that even if validation fails the Scan Data continues through the pipeline with the appropriate parts marked invalid. Each subsequent functional unit can decide what action to take based on the available data. Thus, we preserve the pipeline and decisions can be made in each functional unit independently.

In the second functional block the box ID is found along with box type, scanner unit ID and the vial IDs. The box side label includes a 'box type' so the user does not have to select a type at scan time. The box type defines the layout of the storage locations in the box. The box layout is added to the Scan Data object. The layout of each box type is stored in a configuration file so a new storage box type can be introduced without code changes. Note that the function determines the type of box layout and adds the layout to the data object. The system does not take a separate path for each type of box.

The third block maps each barcode ID to a box storage location based on the pixel position. It may seem that most of the transformation executes in this block but much of the underlying work has been put in place to make the process less complex. The pixel positions of the barcodes are tested against the layout of storage locations created in the previous block and a list is constructed of locations and vial IDs. *Some types of storage boxes such as cardboard hold the vials quite loosely. During the mapping operation vials are given a grade accuracy of the vial positions. The grade provides a confidence factor for vial locations to the database.*

Data accumulates in the Scan Data object as it moves through the process. After the third block the Scan Data object includes raw data from the scanner, all the parsed out IDs, the box definition, box locations with vial IDs, and a running log of the processing in the pipeline. Since the scans are independent, log data is more useful on a per scan basis than as a time based log of all the scans, therefore we keep the log in the Scan Data object. In contrast, scanner communications take place outside the context of a single scan, so that log data is stored in a traditional log file on a time basis.

The next functional unit is rather unique and inherits from the previous generation system (VATS). While 2D barcodes have significant redundancy built-in, they can fail to be read due to frost obscuring sections, water on the label, marks through the barcode, etc. The VATS system implemented a test for unread vials by processing the image from the scan. The scanner sends the server application both an ASCII list of barcodes and a jpeg image of the box being scanned. A storage location is defined by a rectangular subset of the image. The rectangle is tested for a pattern characteristic of the barcode label. Any empty locations meeting the criteria are marked with "decode fail".

The final functional block generates an XML formatted file from the accumulated Scan Data. The file format is a header and box ID followed by the list of storage locations with each location either "empty", "decode-fail", or a vial ID. The XML file also includes the log data accumulated during the processing. A copy of the XML output is stored on the server along with the original ASCII data and image from the scanner unit. By storing a copy of the raw data and processed output and log debug of issues becomes much easier. Also, daily operation and load on the system can be easily tracked.

The database communications block manages the sending of XML output to the laboratory database utilizing a message queuing server. The message queuing server allows a level of decoupling between servers and handles issues like retrying on communication errors. This block exists outside the pipeline flow similar to the scanner communications block. Consequently, it logs to the time based log like the scanner communications block. At startup the connection is established and the connection is torn down on shutdown.

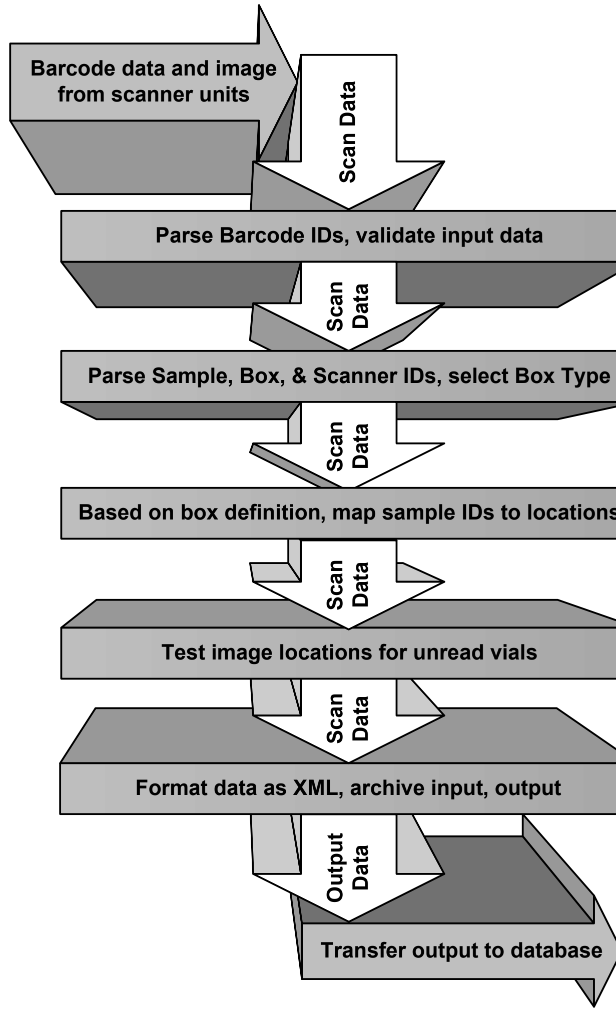

Fig 6. Functional blocks of the server application

VI. Conclusion

Our project was to create a fast and robust device to capture the location of samples stored in a box that allows the laboratories the flexibility to use sample vials of varying dimensions and boxes of varying configurations. The Box Vial Scanner supports sample vials from 1.5 to 4 inches tall held in storage boxes of plastic or cardboard with grid layouts of 9 X 9 or 3 X 8 and can be configured for other layouts. The primary box layout was updated in the field only requiring updating a configuration file on the server. The boxes are captured and processed in less than 2 seconds with most of the time required for image transfer from the device. The system is currently deployed in a high volume laboratory where the users consider the unit 'critical' and typically resort to hand scanners only for single vial changes. Units are in preproduction testing in other laboratories.