EXSCALABAR SOFTWARE SPECIFICATIONS

PURPOSE

The purpose of the document is to define the software requirements from a user standpoint. The requirements defined in the following sections are not intended to define implementation, they simply outline the expected functionality of the software product. The functionality and features defined below will be in the final software package. This document defines the minimum functionality that will be provided. It is expected that throughout the process the software specification will be modified. Only prior to completion of the development of the software will the specifications be considered complete.

HIGH LEVEL ARCHITECTURE

The general system architecture is shown as a deployment diagram in Figure 1. This figure shows the generic system layout and provides a high level view of the architecture. The DAQ software will be deployed on an embedded controller and will communicate with various hardware devices via serial connections or cards connected to the controller via a backplane. A startup executable will run when the system is powered up and will consist of VIs compiled into a general startup.rtexe. In addition, Web Services will be provided for communication via http (if desired). Not all of the components defined in Figure 1 are guaranteed for final delivery (such as the

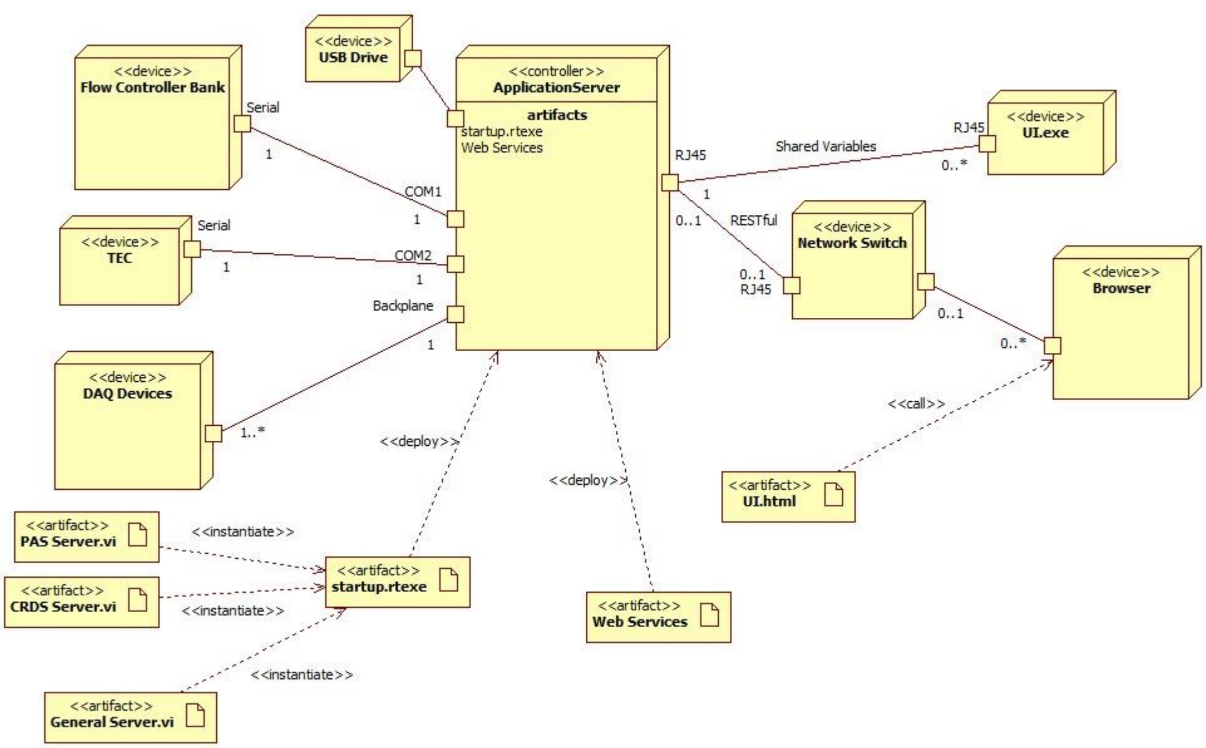

Figure 1. Deployment diagram for the EXSCALABAR instrument. The deployment diagram shows the general relations between physical entities (≪device≫) and the software used by those entities (≪artifact).

UI.html software) – the diagram is only intended to provide an overview of the main functionality and how the software will interact with hardware entities as well as a user interface (UI).

GENERAL OPERATION

The two main subsystems, the CRDS and PAS, will run asynchronously. Each of the subsystems will be based off of software previously developed for the NOAA AOP instrument. The base software is provided to the UK Met Office by NOAA at no cost. Engineers at NOAA will provide no support for this software.

The base software for both the PAS and CRDS are built off of something I call the Instrument Framework and therefore have similar attributes. Both are built off of an abstract class called *Instrument.lvclass*. This class will provide the basic functionality that is to be defined by the children. The functionality sketched out by the framework is defined in the operations of the Instrument class in the UML diagram in Figure 2. The actual structure will be considerably more complicated and much of the details have been omitted to demonstrate the simplicity of the framework. Functionality specific to the EXSCALABAR instrument will be developed in children of the PAS and CRDS instruments defined below.

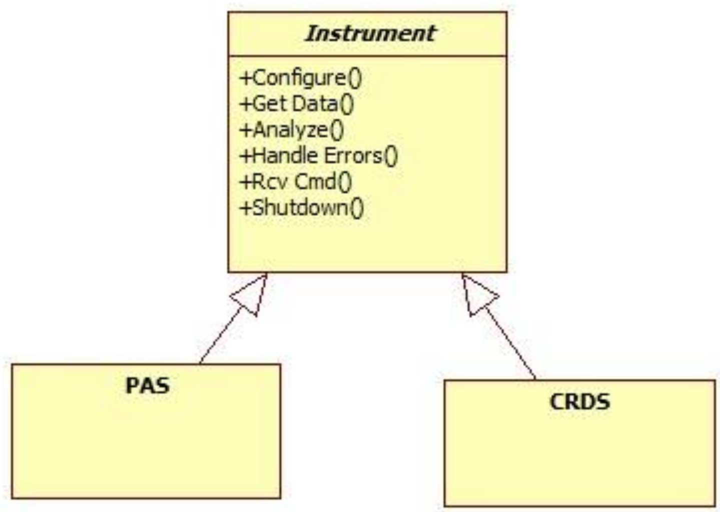

Figure 2. Basic class structure for the subsystems PAS and CRDS. This diagram is far from complete but only intended to demonstrate how functionality is shared between the two main subsystems. The code represented above provides the general framework for the EXSCALABAR instrument and is provided by NOAA free of charge. The EXSCALABAR software will build on this framework by extending the classes above to provide specific functionality.

The flow of operation and how it pertains to the actual operation of both subsystems is shown in Figure 3. In each subsystem, the instrument will be configured. If configuration fails, the system will immediately proceed to shut down. EXSCALABAR software will provide indication either through bits set or through a UI that the subsystem is no longer functional. If configuration succeeds, the software will proceed to get and analyze data. These processes occur in parallel to prevent missing data acquisition periods due to latency in the processing. The implication of this is that data that is analyzed is one cycle behind the data being retrieved (expect to see a 1 s delay between the current time and the acquired data time). This is termed pipelining and is intended to take advantage of the multicore architecture of the hardware.

When the data is analyzed, the relevant analyzed data is merged with most recently retrieved data and the analyzed data is shipped for consumption by subscribers (user interfaces, files, etc.). If an error is thrown during either of these processes, the error will be handled appropriately following the data acquisition cycle.

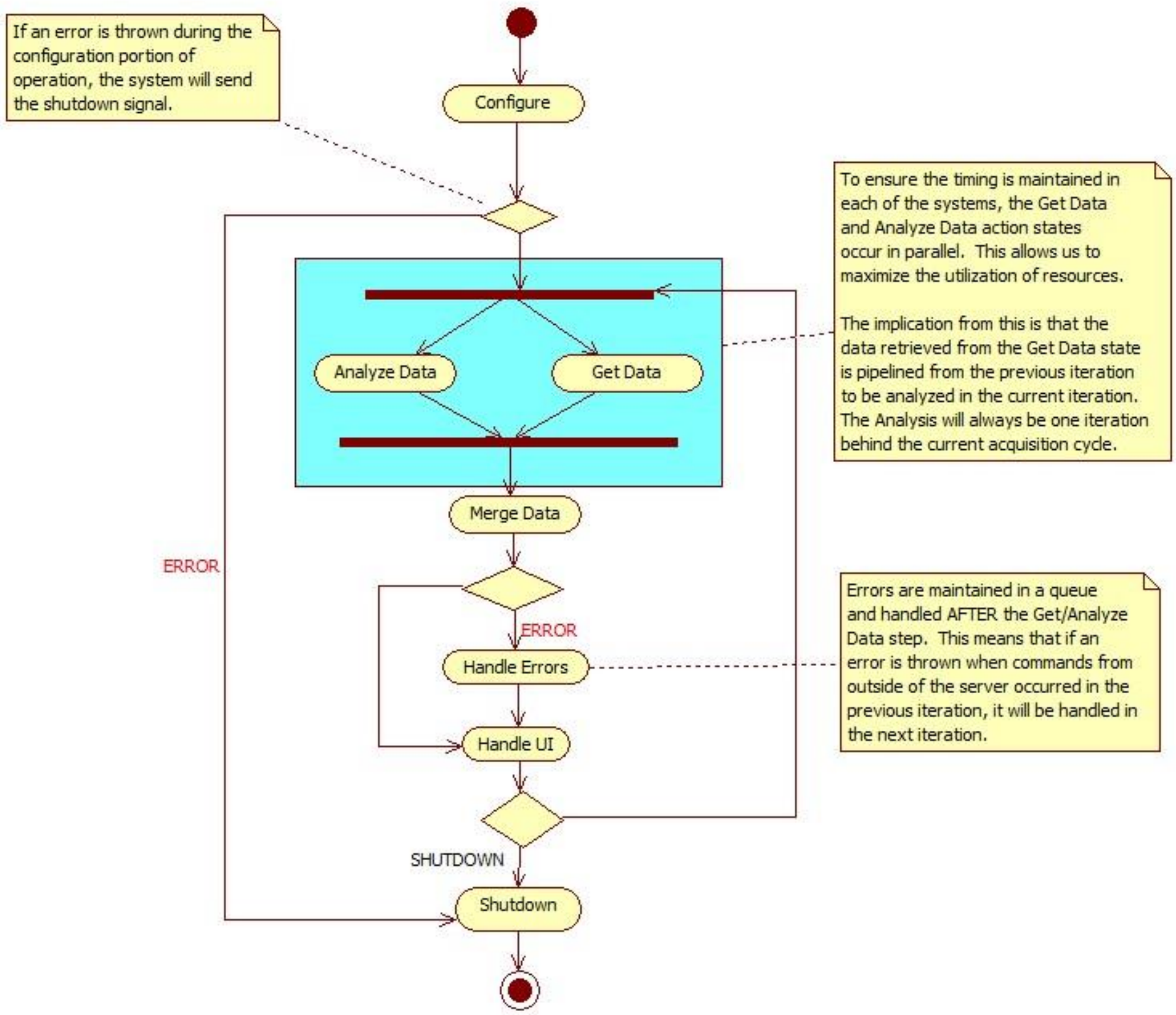

Figure 3. This diagram shows the general flow of operations for both the PAS and CRDS subsystems. The flow is to some extent dictated by the INSTRUMENT framework provided by NOAA AOP and provides for discrete changes without widespread code restructuring.

CALIBRATION

The two main subsystems require regular calibration to establish baselines and backgrounds. Regular zero backgrounds provide information on the influence of noise. Ozone calibrations are used to provide a correlation between the extinction measured by the CRDS and the absorption by the PAS. The pressure calibration provides a calibration for the dependence on PAS cavity characteristics on pressure. During any of the calibration periods detailed below, the regular flow of operation shown in Figure 3 is not disrupted. Rather, in some cases, there may be additional files produced as stated below and described in subsequent sections. Data collected during calibrations will generally not be considered publishable – this data is collected for the proper operation of the EXSCALABAR instrument and is used to aid in the producing publishable data.

ZERO CALIBRATION

The software will provide the ability to regularly provide background data. This data provides key information for determining baseline signals. During background checks, the main system flow will be directed through a filter and data will be acquired for a user defined period of time.

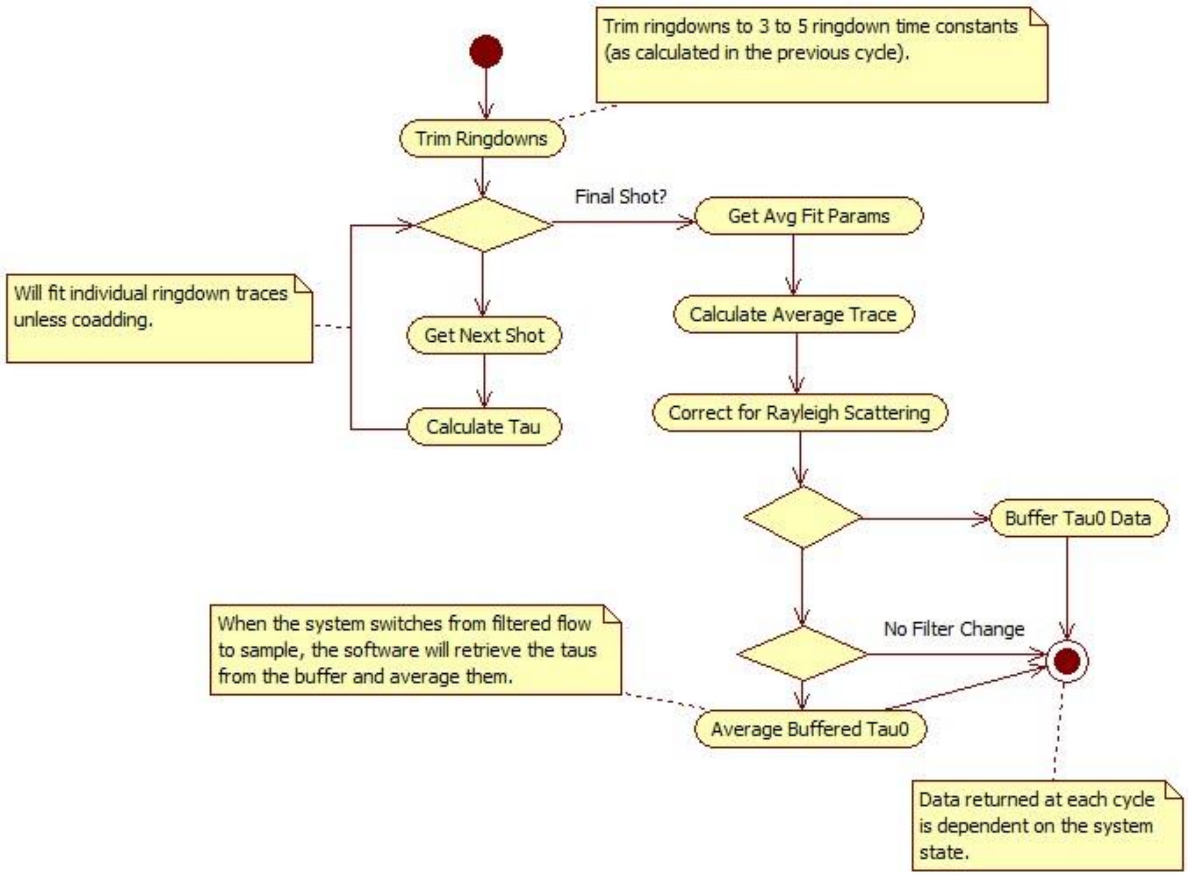

Figure 4. General operational flow for the CRDS data analysis. See Figure 5 for how the CRDS data analysis fits into the general flow of the CRDS data acquisition system.

The PAS and CRDS will handle the zero data differently. As shown in Figure 4, the CRDS will use the data collected on filter to produce a τ_0 which in turn will be used to calculate extinction. In this case, the CRDS will actually average τ's in a buffer when the filter is bypassed (i.e. the τ_0 is an average of the data retrieved over the filter period). The PAS, as shown in Figure 6, will use the zero data to remove background from the baseline microphone signal.

SPEAKER CALIBRATION

The speaker calibration in the PAS measures the resonant frequency f_0 and the quality of the resonance Q. During speaker operation, the speaker is sent a symmetric chirp as shown in Figure 5. The software will provide access to parameters for determining the shape of the chirp (i.e. the range of frequencies probed by the chirp) as well as the period of the calibration and the length of the calibration. The chirp will be adjustable at run time – this means that the chirp is not static but different regions may be probed in the frequency spectrum.

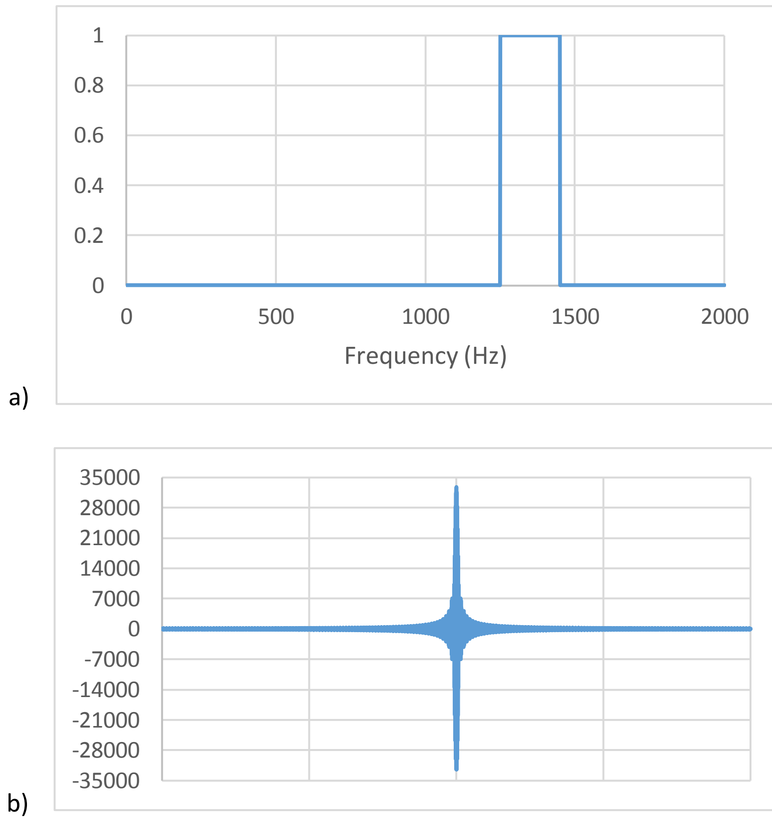

Figure 5. The image above shows a signal generated for speaker calibration. a) Representation of the "chirp" in the frequency domain. B) Resulting signal in the time domain sent to the speaker. The resulting signal measured by the microphone will differ from this as it will be affected by the resonance of the cavity.

Figure 6 shows how the analysis of the speaker fits into the general data analysis operation for the PAS as outlined in Figure 3. During speaker calibrations, the resonant frequency f_0 and the quality of the cell resonance Q are calculated based on fit of the frequency distribution data produced by the microphone during the chirp. The spectrum produced approximates a Lorentizian distribution. The distribution is fit using the Levenberg-Marquardt algorithm, a non-linear least squares method. In the next cycle, the PAS subsystem takes advantage of these new parameters by operating the laser at the new resonant frequency of the cavity.

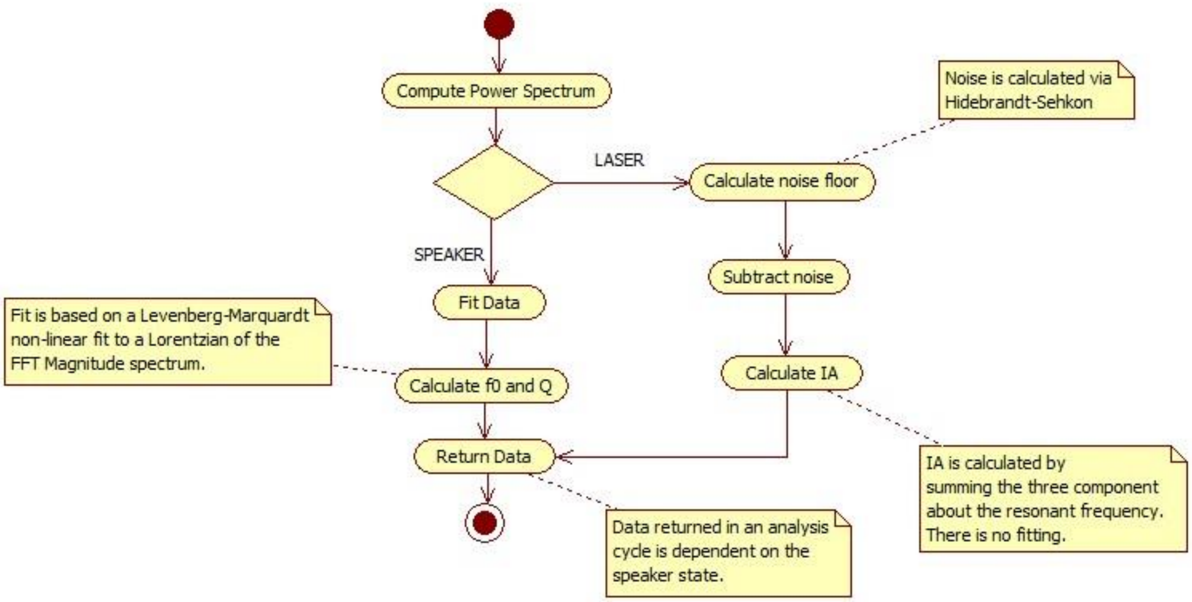

Figure 6. Activity diagram for analysis of PAS data (see Figure 3 for how the analysis fits into the general flow of operation).

OZONE CALIBRATION

Ozone calibrations allow the extinction measured by the CRDS to be correlated with the absorption of the PAS and are crucial for converting the integrated area measured by the PAS into a physical quantity. zone calibrations typically are performed at the start of operation and at the end. An example of a typical ozone calibration sequence is shown in Figure 7.

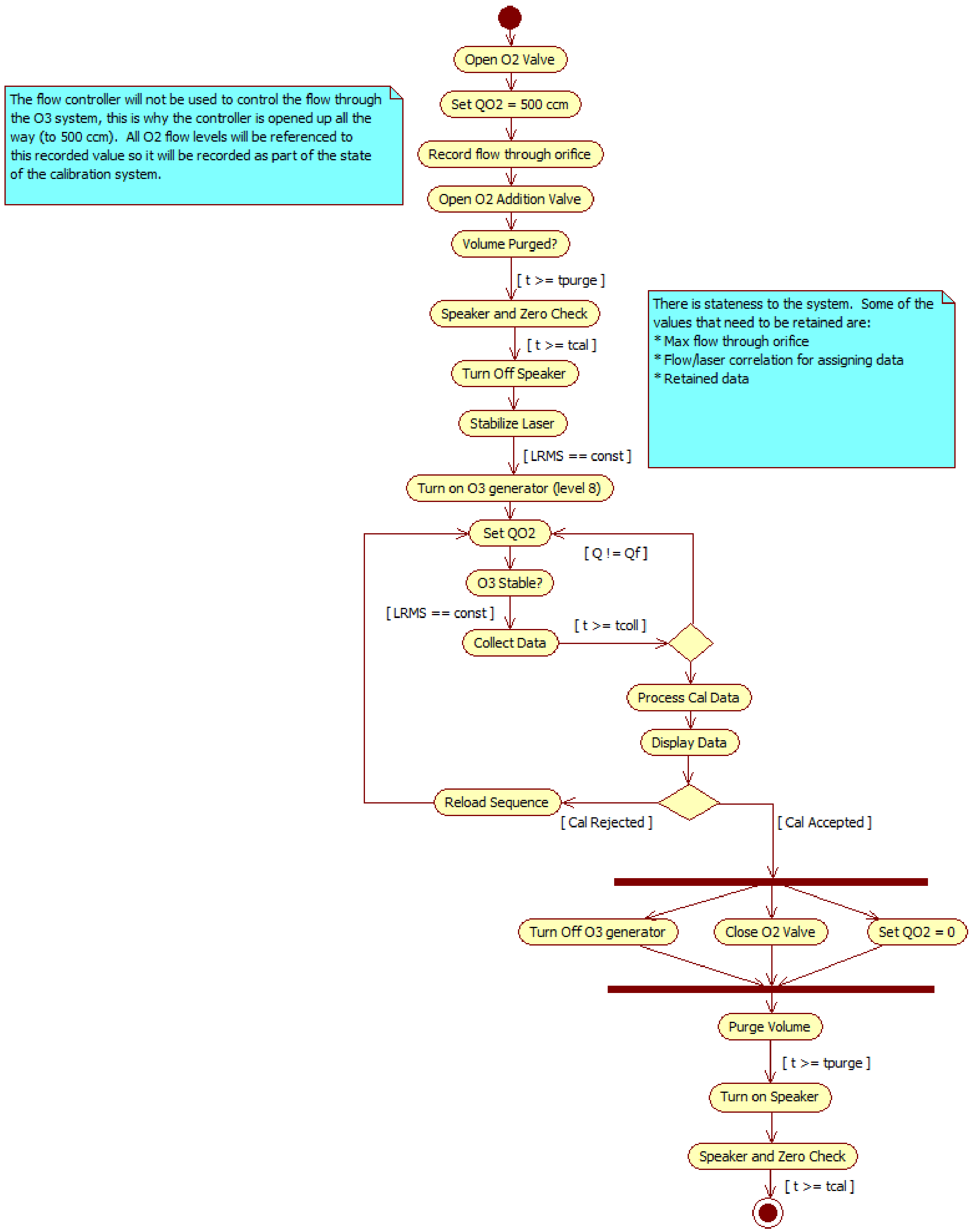

Figure 7. An example of an ozone calibration sequence.

The data acquisition software will provide availability of parameters necessary for conducting ozone calibrations through both a configuration file (discussed below) as well as through the user interface. These parameters include but are not limited to:

- An array of flows for controlling O_3 concentrations.
- Various times required for operation of the calibration (i.e. purge, filtering, speaker calibrations, data collection, etc).

Data collected during this calibration will be copied to a separate file. The expected output parameters (slope and intercept for the relation between integrated area and absorption) will be applied immediately to the extinction recorded in the data file and that displayed by the user interface.

PRESSURE CALIBRATION

The ability to perform well characterized, sequenced pressure calibrations will be provided. The data acquired during these periods will provide information similar to what is provided during typical speaker and zero measurements, but will allow the users to extrapolate to pressures not directly measured during these calibration periods. The general flow of a pressure calibration is shown in Figure 8. In this diagram, the laser response to pressure is characterized

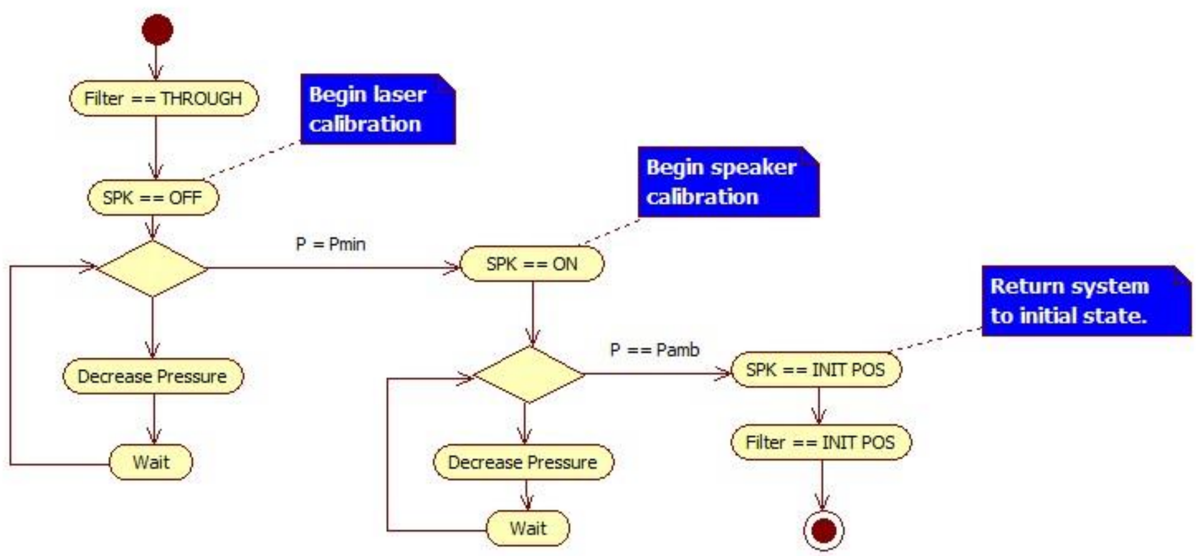

Figure 8. Activity diagram for a pressure calibration. Calibration for the PAS cell proceeds during both the decrease and increase of pressure. In this diagram, the laser calibration occurs as pressure decreases and the speaker occurs as the pressure increases. The system is returned to the final state upon exit.

For pressure calibrations, the user interface and the configuration file will provide the ability to adjust the following parameters:

- Minimum pressure
- Number of pressure steps
- Period to wait at each step

Because pressure calibrations are not intended to run at regular intervals, the initiation of pressure calibrations will require user intervention, either through a physical switch (such as a push button which will load the configuration and run it) or through the user interface. During pressure calibrations, data will be copied to a separate file as described below.

MODES

SPEAKER AND ZERO CALIBRATIONS

For zeroing and speaker calibrations, the software will provide controls for adjusting the frequency and length of these calibrations. These calibrations will be performed at regular intervals. In addition, the software will provide the control to "link" the two calibrations. That is, the speaker calibration (the shorter of the two) may be set to proceed at some point in the zero operation.

OZONE CALIBRATIONS

The software will allow the user to specify whether the calibration is to proceed manually or at regular, user defined intervals.

RH CONTROL

The EXSCALABAR data acquisition software will provide the capability to operate two humidified channels on the CRDS. Both will have the capability to operate at fixed temperature as well as in a scanning mode. The details of the implementation will be hardware dependent and therefore are not currently specified. As of the time of the writing of this document, the only parameters considered configurable for this are the desired RH setpoints. This may change to include actual control parameters for each, but as stated before, this will be left to the final hardware specification.

FILES

A list of possible files used by the instrument follows. Files that are generated by the system at run time (such as a copy of the configuration file, the data files, logs, etc.) will be written to a common directory. The main directory will reside on the controller system or a hard drive attached and will be specified by the user in the configuration file. For each power cycle, the system will generate a new directory that will be stamped with the date of the power up if that file does not exist. Files below will be stored in this folder.

STORAGE

All files will be saved to the controller or a disk connected directly to the controller. Data can be viewed on the user interface in real time but will not be written to the device hosting the user interface.

INITIAL STORAGE DEVICE STATUS

On startup, the software will provide the status of the device including whether it is present and the amount of space that remains on the device.

STORAGE DEVICE FAULT STATUS

Faults may occur for several reasons: device is full, device is disconnected or device has failed. Upon fault, user will be notified of storage faults and prompted to repair issue.

CONFIGURATION FILE

The configuration file is intended to provide the user access to input parameters at instrument startup. The final product will include a configuration file that will properly initialize the instrument.

SOFTWARE CONFIGURATION

The software configuration file and the corresponding parameters are intended to provide a method for initializing control parameters available to the user via a user interface (when present). These values are expected to regularly change.

FORMAT

The configuration file will be a standard INI file that can be edited in any text editor. Logical key-value groups will be stored in sections.

MODIFYING THE CONFIGURATION FILE

The format of the configuration file is such that it may be manipulated directly via a text editor. It is important to note that direct manipulation of the text file may result in unexpected system startup configurations. Do not let 4 year olds or blundering post-docs directly interact with the configuration file. Both have been known to ignite previously believed inflammable objects. An editor will be provided for fool-proof manipulation of the configuration data.

CONFIGURATION FILE DATA

This section describes the data that will be stored in the configuration file. The values within the configuration file are defined in the following table.

Table 1. Configuration file data for the EXSCALABAR instrument.

Section	Key	Value	Description
Pressure	N_{steps}	Numeric	Number of steps to take for pressure calibration.
	P_{min}	Numeric	Minimum pressure for pressure calibration
	t_{wait}	Numeric	Time in seconds to wait at each pressure step.
Ozone	O_3 Power	Boolean	Initial power state of the ozone lamp
	f_{O3}	numeric	Number of sampling cycles between O_3 calibrations.
	$n_{cal\ levels}$	Numeric Array	Array of pressure levels for O_3 calibrations.
	Q_{O3}	Numeric Array	Array of flow rates for ozone calibrations.
	t_{wait}	Numeric	Wait period for changes in O_3 concentrations due to flow or pressure changes.
	t_{acq}	Numeric	Acquisition times for O_3 calibrations.

	Cal Mode	Boolean	Calibration mode. TRUE = automatic, FALSE = manual.
	$t_{warm\ up}$	Numeric	Warm up time for lamps prior to calibration.
MFC	Q_0	Numeric Array	Initial flow rate for each controller
	Port	String	Communication port for mass flow controller bank.
Filter	Filter Pos	Boolean	Initial filter valve position. TRUE = filter, FALSE = sample.
	Filter mode	Boolean	Filter cycle setting. TRUE = automatic cycle, FALSE = manual cycle.
	t_{filter}	Numeric	The amount of time on filter in an autocycle in seconds.
	t_{sample}	Numeric	The amount of time on sample while autocycling in seconds.
CRDS	f_{laser}	Numeric Array	Initial laser repetition rate in Hz for each of the lasers.
	DC_{laser}	Numeric Array	Duty cylces defined for each of the lasers.
	Coadd	Numeric Array	Number of shots to coadd for each of the channels.
	λ	Numeric Array	Laser wavelengths for each cell.
	n_{cells}	Numeric	Number of CRDS cells present in the system.
	RH	Numeric Array	Initial RH setpoints for channel humidifier.
	l_{cell}	Numeric Array	Optical length of cells.
PAS	V_{speak}	Numeric Array	Voltage limits for each of the speakers.
	n_{cells}	Numeric	Number of cells in system.
	m	Numeric Array	Initial calibration slope for converting integrated area to absorption for each cell.
	b	Numeric Array	Initial calibration offset for converting integrated area to absorption for each cell.
	T_{cell}	Numeric Array	Initial cell temperature specified for TECs.
	$T_{chiller}$	Numeric	Initial temperature of large thermo electric cooler.
	Laser Power	Boolean Array	Initial power up state of all lasers.
	TEC Power	Boolean Array	Initial power up state of cell TECs.
File	f	Numeric	Rate at which to save data to file in Hz.
	folder	String	Name of top level folder on data drive to which data will be saved.
	Write?	Boolean	Initial setting for writing data to file.
General	Solenoid positions	Boolean	Initial solenoid valve positions. TRUE = open, FALSE = closed.
	Msg Log Details	String	Provides the ability to turn off or on different streams generated by the system. Possible values include: ALL, CRDS, PAS, ERROR, WARNING, NONE. The value will accept multiple, comma delimited entries.

CONFIGURATION FILE STORAGE

The configuration file will be stored on the main hard drive of the data acquisition system under c:\cfg. This file may be accessed via FTP.

HARDWARE CONFIGURATION

The hardware configuration allows the user to properly configure the hardware for communication with the software. The configuration will be conducted using the National Instruments Measurement and Automation Explorer (MAX). The configuration consists of

- Names of the hardware present in the system
- Tasks utilizing the named hardware which incorporate channels on the card. Tasks will contain information such as sampling rate and the terminal configuration.
- Scales used by the individual channels in each task to provide useful information. This might be something such as scaling a 0 to 5 V signal to provide a 0 to 100 output.

This configuration is expected to change rarely if ever. The names within the tasks and in the hardware are used to communicate directly with the data acquisition software. It is very important to note that if you don't know what you are doing, stay away from this file! Unless of course you enjoy calling your consultant. Then, by all means….

For those times where the configuration implodes (as when it is handled by the monkey above), it is recommended that the file be periodically exported via the MAX interface and backed up. The file that is produced by this process will have a .nce extension and can be used to restore the configuration to its previous state.

STANDARD DATA FILE

The instrument shall produce a set of data files at a user defined rate. Since the PAS and CRDS run asynchronously, each subsystem will have a standard data product. These product will be recorded in text (ASCII) format that will be tab delimited.

FILE NAMES

Each file will be stored in a user defined folder (as described in the previous section) and will be stamped with the date, time and the subsystem. The name format will be yyyymmddhhmmss.txt where **yyyymmddhhmmss** is the date and time the file was started.

AUXILIARY DATA FILES

In addition to the files discussed above, there are several other files that will be saved periodically.

CALIBRATION FILES

During ozone calibrations, in addition to the standard data file defined above, a separate data file will be written to containing all of the information found in the above files. The file will contain a header which will provide specific information relating to the calibration. This information will include the date and time of the beginning of the calibration, the various time used in the calibration and the flow rates used for the different calibration levels.

GRANULAR CAVITY RINGDOWN DATA

As described above, this mode will allow the user to capture a set amount of ringdowns for offline analysis. During this mode, the data written to the standard data file (defined above) will contain housekeeping data but no CRDS analysis (i.e. τ, extinction, etc). This will be done to limit the amount of processing required by the system as it writes large chunks of data to the file. The file will be written in binary format and will contain an array ringdowns and the time associated with each array of ringdowns (usually acquired at 1 s intervals).

SYSTEM MESSAGE LOG

For each operation, a system log file will be generated. This file will contain information regarding the operation of the instrument and is useful for debugging or forensic purposes (i.e. determining what user action screwed up the experiment so badly). This information will regard such actions as

- Aperiodic user generated commands.
- Periodic changes in mode
- Errors and warnings generated by the system.
- Unexpected changes of state (such as shutdowns due to error).
- Current modes with respect to calibration or zeroing.

The system log will provide functionality to turn the verbosity on or off, ignore codes report only messages of a certain type, etc. This functionality will be set through the configuration file above.

Each message will be time stamped and will contain an indication of what subsystem generated the message (i.e. CRDS, PAS, etc). The messages will look something like the following:

> *16:22:50[pas] User requested change of state.*

Errors generated will have a similar stamp but will be proceeded with by an indicator such as *[ERROR]*. Logs will be stored in the same directory as the current data files.

USER INTERFACE

This section describes the specification for the user interface. Since this is likely to vary during the development process, no attempt has been made to make an exhaustive list of all the capabilities and displays nor has any attempt been made to define the execution platform. This section is expected to change throughout the design process. The customer is encouraged to add as necessary functionality desired for the manual operation of the instrument.

CONTROL

The user interface will provide access to all configurable data outlined in Table 1. In addition, the user interface will provide the ability to change

- The filter state.
- The data recording state.
- The time on the chassis.
- The calibration state (both the mode and the current operational state).

UPDATES

In standard operational mode, the user interface will be updated at a rate of 1 Hz. During alignments, the user will have the opportunity to change the rate of update. The rate to which the value may be changed is currently not determined.

GENERAL DISPLAY

The user interface will provide access to all data acquired either via discrete displays (such as numeric indicators) or through plots. Expected plots include those for housekeeping and the main data products for each of the instruments (such as extinction, τ, integrated area, resonant frequency, etc.).

NETWORK COMMUNICATION

The embedded system will have the ability to communicate with an array of devices within a single subnet. Data may be broadcast across a single Ethernet cable or may be provided over a wireless router.

STANDARD COMMUNICATION

Standard communication will be provided via a library of shared variables. These variables will have access to the subsystem controls as defined above as well as data produced by each subsystem. \par

WEB SERVICES

The chassis will host web services with appropriate http headers for cross origin requests (i.e. so that these services may be accessed asynchronously for html updates). These services will retrieve their data from the data server using the shared variables as defined above. The web services will provide information using JavaScript Object Notation (JSON).

DOCUMENTATION

Documentation of the system will proceed at several different levels. A manual will be provided that documents code functionality. In addition or as part of the same manual, a user manual will be provided that will give a high level view of the functionality of the system and will provide a how-to guide for the user. The format of the help documentation will be HTML unless otherwise specified. HTML is platform independent and can be loaded properly on a variety of devices.

SOFTWARE DOCUMENTATION

As part of the final package, the data acquisition software will contain detailed information about the functionality of the code. This functionality will be documented within the Vis themselves as well as in the documentation window of the VI (this is the window that pops up when a user hovers over a particular VI and has the help functionality turned on). In addition, this documentation for each VI will be compiled into a single help file along with more general information regarding architecture and broad functionality (such as that information given at the beginning of this specification).

USER MANUAL

In addition to the software documentation, the UK Met Office will be provided a general user manual. The user manual will describe the actual possible user interactions with the system that are not related to software development. This will include but is not limited to:

- Editing and loading of configuration files.
- Starting the software and connecting to the instrument via a user interface.
- Editing and changing calibration modes in real time.
- A general description of data displayed on the user interface.
- A description of the data saved to file and how it is saved to file.

SOFTWARE VERSION CONTROL

Version control of all software products will be maintained using Subversion on the MSR Consulting website. The repository can be found at http://msrconsults.com/exscalabar_trac/ under Browse Source. The repository will be divided into three sections: trunk, branch and tags. Trunk and branch folders will contain software considered to be under development. Trunk will be the main repository path as developed by MSR Consulting. If engineers at the UK Met Office wish to work on a particular version, they are encouraged to develop branches which will be checked in and out of the brank folder. These branches will be merged with the trunk as development proceeds.

The tags folder represent release code. The code is considered to be ready for use and can be checked out for operation of the instrument. Tags will generally be logged in folders either with version or the date the code was tagged. Engineers who wish to test code but not alter it are encouraged to check out the code from this repository.

TRACKING BUGS AND CHANGE REQUESTS

Bugs and change requests will be submitted through the Trac site for this project (see link above). For directions on how to use the site or if changes to the ticketing process are required, please contact on MSR Consulting.

LABVIEW VERSION

Most software in this product will be developed in National Instrument's LabVIEW. As this project will straddle LabVIEW release periods, it is likely that code at the outset will be written in 2013 but will be completed in LabVIEW 2014. This will extend the lifetime of the code and hopefully aid in the maintainability of the code.