MANAGING NEUTRON BEAM SCANS AT THE CANADIAN NEUTRON BEAM CENTRE

Mark Vigder, National Research Council, Canada
Martin Lee Cusick, Canadian Nuclear Laboratories, Canada
David Dean, National Research Council, Canada

Abstract

The Canadian Neutron Beam Centre (CNBC) of the Canadian Nuclear Laboratories (CNL) operate six beam lines for material research. A single beam line experiment requires scientists to acquire data as a sequence of scans that involves data acquisition at many points varying sample positions, samples, wavelength, sample environment, etc. The points at which measurements must be taken can number in the thousands with scans or their variations, having to be run multiple times. At the CNBC we have developed an approach to allow scientists to specify and manage their scans using a set of tools we have developed. Scans are specified using a set of constructors and a 'scan algebra' that allows scans to be combined using a set of scan operators. Using the operators of the algebra, long complex scan sequences can be constructed by combining shorter simpler scans. Based on the constructors and the algebra, tools are provided to scientists to build, organize and execute their scans. These tools can take the form of scripting languages, spreadsheets, or databases. This scanning technique is currently in use at CNL, and has been implemented in Python on an EPICS based control system..

INTRODUCTION

The Canadian Neutron Beam Centre (CNBC) conducts material science experiments using neutron scattering techniques. The neutron source is an experimental reactor that provides six beam lines for neutron scattering. Each beam line is specialized for a particular scanning purpose, such as triple-axis spectrometry, reflectometry, stress-scanning diffractometers, etc. In addition to the equipment controlling the spectrometer beam lines, a set of ancillary equipment is used to control the sample environment. This includes equipment such as furnaces, cryostats, stress rigs, etc.

When conducting a single experiment, scientists have a very large number of independent variables they must control. For the beam line this includes beam position and focus, energy levels on the main and diffract beams, and polarization of the main and diffract beams. Sample location and orientation must be controlled, as well as sample environment (temperature, stress, etc.) Setting each of these independent variables could involve setting multiple sub-variables, or sequencing through a number of steps to correctly set the variable.

An experiment involves setting the independent variables to the appropriate values and initiating a data acquisition phase for a specified duration. The independent variables are then reset to new values and the data acquisition repeated. A single experiment repeats this process performing a data acquisition at possibly thousands of different points within the n-dimensional space formed by the independent variables. Acquiring the data at a sequence of points in the experiment space is referred to as scanning.

Scanning is often a dynamic and interactive process. When conducting scans, scientists monitor the data as it is being acquired and will often stop, restart, modify, rerun, or resequence a scan. Creating and managing these scans and dynamically interacting with the scanning process presents a number of challenges to the scientist and to the system developer who must support the scientists work.

OBJECTIVE

The objective of this work is to develop a method that allows scientists to easily create, manage, and execute their scans. The work was undertaken during a major upgrade of the control and data acquisition system. The original spectrometer control system at NRU was developed incrementally over more than 30 years of work and ran on a VAX based system. In the course of replacing it with a distributed Linux based system, the opportunity arose to replace the outdated user interface with a modern interface, including improving the techniques for scientists to perform the scanning for experiments.

In redoing the scanning control and management, the following requirements were followed.

First, there was not to be a major paradigm shift in how the scanning process was managed by the scientists. The old system was dated, had no underlying model, and carried thirty years of baggage in its design and software. However, it also was designed by the users who knew what they needed, was very efficient and easy to learn, and carried thirty years of detailed knowledge of the scanning process. The new system was not to be a major shift in the process, but rather was to provide a better model of scanning, a refactoring of the design and implementation, and much more capability and flexibility that would be provided to users.

Second is the requirement that identical scanning control management software is to be used on all the beam lines. With minimal resources to maintain and manage the software, having different software on the different beam lines was not a viable option. Even software variants, no matter how similar, create software management problems. However because each beam line is unique this required that the system
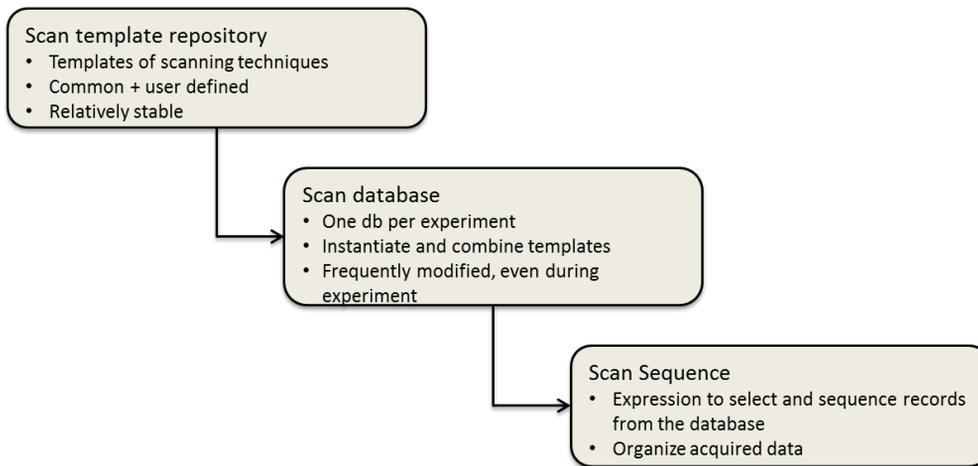
Figure 1: Managing scans.

be designed in such a way that it allowed for different configurations for different beam lines.

Finally, the system must be 'easy' to learn for incoming scientists. The CNBC is a user facility that hosts a number of researchers for days or weeks at a time. A local contact guides these users through the process of conducting an experiment. However, the visiting scientists should be able to work independently and not require the local contact for relatively routine operations performed during the experiment. A system in which the full functionality of the system can be learned by a naïve user within a few hours is clearly impossible. However, many of the operations performed by scientists during an experiment are simple in concept and repetitive in nature, and these should be available to visiting scientists with minimal training. More complex operations would still require intervention from the local contact.

MODELLING SCANS

The first step in upgrading the scan management and control system was to build a more formal model of scans. This was done using a simple algebra. Working within the n-dimensional space of the independent variables, the following can be defined.

An *independent variable* is a name and a type. The type is the set of valid values for the variable.

A *scan point* is a set of tuples, were each tuple consists of an independent variable, and an optional value for the variable.

A *scan* is an ordered sequence of scan points such that each point of the scan contains the same set of independent variables.

Scan Operators

As part of the scan algebra, a set of binary operators are defined. These operators are used to construct complex scans from simpler scans. The binary operators used are the following.

Sequencing (**+**). The scan sequence operator extends one scan with a second scan, **S1+S2** creates a single scan with the ordered scan points of **S1** followed by the ordered scan points of **S2**.

Looping (*****). The scan looping operator allows one scan to be repeated for every point of another scan. The expression **S1*S2** is a scan of length **len(S1)*len(S2)** in which each point of **S1** is replaced by a scan consisting of that point extended with the entire scan **S2**. For example, if scan **Q1** steps the scattering angle **Q** through a sequence of points, and scan **T1** steps temperature through a sequence of values, then **Q1*T1** is a scan that scans the scattering angle **Q** at each temperature in **T1**.

Interleaving (**|**). The scan interleaving operator interleaves the scan points from two scans. For example, if **Q1** and **T1** are defined as in the previous paragraph, and scan **BG1** offsets an angle to do a background measurement, then **(Q1 | BG1) *T1** scans **Q** at each temperature of **T1** while also performing a background count for each point.

Dot product (**^**). The dot product operator joins two scans together point by point.

MANAGING SCANS

The method for managing scans is illustrated in Figure 1. There are three steps to the process.

Scan repository

The first step is the construction of a *scan repository*. The repository contains parameterized versions of standard scans. The most basic parameterized scans in the repository include the following:

- **step**. The step scan steps a set of independent variables through a linear set of equidistant steps. For example, step angle phi from 23 to 29 in steps of 0.5 degrees.
- **points**. A point scan assigns a sequence of values to a set of independent variables. For example, it can be used to set the main beam and diffract beam flipper coils in order to sequence through the different polarization options.
- **setup**. A setup scan is a single point scan. It is used to simply set a set of independent variables to

specific values. It is usually combined with a **step** scan to initialize a set of values and then step an independent variable.

In addition to these parameterized scan templates, a number of more complex scan templates are also stored in the repository. For example, the **texture** scan stores over a thousand scan points representing different sample orientations.

The scan repository is relatively stable and will generally contain no more than a few tens of scans at most. Additions to this repository are generally required only as new scanning techniques are developed.

Scan Database

A scan database is created by the scientist for each experiment being conducted. Each record of the database is a scan and is constructed by instantiating scan templates from the repository and combining the instantiated scans using the scan operators.

Unlike the scan repository, the scan database is highly dynamic. The scientist initially builds the database before conducting the experiment and is continuously modifying and updating the database during the course of the experiment.

Each record of the database is constructed by the user by first selecting the templates from the repository and instantiating them. For example, to do a texture scan across a range of temperatures, the user instantiates a texture scan and a step scan stepping temperature and combines them using the looping operator.

Sequencing

Once the user has constructed the scan database, the user executes the scans by creating a sequence expression. A sequence expression extracts records from the scan database and specifies the order in which the records will be executed. A few simple operators are available for representing repetition, sequencing, and selecting a range of records. For example, given that records within the database are referenced by record number, the expression: **(3-10), 25*10, (10-3)** instructs the system to execute scan records 3 to 10, followed by executing record 25 ten times, followed by executing records 10 to 3.

IMPLEMENTATION

The scanning control and management system has been implemented and is currently being deployed on the various beam lines at the CNBC. The main elements of the implementation include the following.

The scan algebra is implemented using the Python dynamic language. A scan class is defined that encapsulates the scan properties. Being a dynamic language, operators of the language can be overloaded allowing Python expressions to be used in a natural way to write the scan algebra expressions.

The repository is simply a set of parameterized Python functions. These functions are called *scan constructors*. Instantiating and calling the function returns a scan. Being relatively small in number, no special database is required to organize the scan constructors. Moreover, with a little Python knowledge and understanding of the scan operators, users can easily write their own scan constructors.

The database required a tool by which the scientist could construct scan records by calling scan constructors and combining the scans. It is possible for records to be constructed directly in Python and the resulting scans stored in a database. However it was felt that this would add a level of complexity that would be a barrier to many of the users. Most experiments are very regular and repetitive in how they scan, and a relatively easy solution is to represent the scan database as a spreadsheet, with the columns representing the scan templates and their parameters, and each row representing a scan record. This provides the user with the full editing capability of the spreadsheet for constructing the scans, while providing an intuitive representation understood by most scientists.

Finally, a GUI is provided through which users can view scans, construct sequences, execute and monitor scans.

DISCUSSION

The old scan management system developed over many years, and which has proven effective, has been used as a basis for the major upgrade that has been performed. This has resulted in a system which is familiar to users but which also has the power and capability to be further evolved. Since it was a complete rewrite of the code, we were able to start with a clean slate and hopefully will be as successful over the next thirty years with the new system.

The formalization of the concept of scanning using the simple algebraic techniques has proven to be a useful technique with both simplicity and power. Complex scans of thousands of points have been constructed using the algebra.

Python, always popular within the scientific community, provides an implementation of the algebra that is easily understood by both developers and scientists. Using spreadsheets for representing the scan database was a technique that scientists were very comfortable with. It is limited in its capability but does handle the majority of requirements in an intuitive manner.

The system is currently being deployed at three of the beam lines at the CNBC, with plans to deploy on the other three lines.