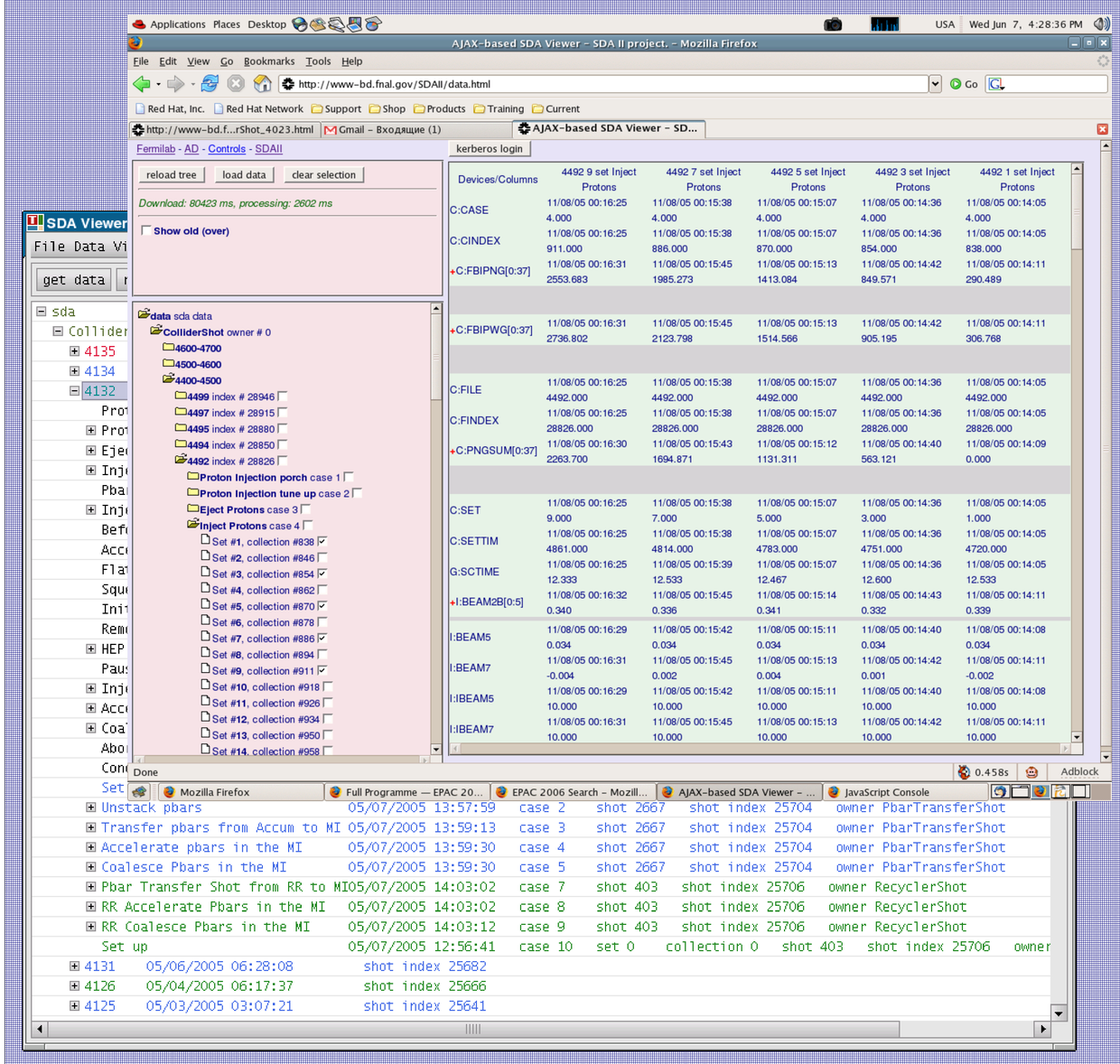
Portable SDA (Sequenced Data Acquisition) with a Native XML Database

Timofei B. Bolshakov, Elliott McCrory, Fermilab, Batavia, IL, U.S.A.

Funding Agency: Fermilab is operated by Universities Research Association, Inc., under Contract No. DE-AC02-76CH03000 with the US Department of Energy.

Abstract.
SDA is a general logging system for a repeated, complex process. It has been used as one of the main logging facility for the Tevatron Collider during Run II. It creates a time abstraction in terms understood by everyone and allows for common time tick across different subsystems. In this article we discuss a plan to re-implement this highly successful FNAL system in a more general way so it can be used elsewhere. Latest technologies, namely a native XML database and AJAX, are used in the project and discussed in the presentation.

Shot structure in SDA Viewer.

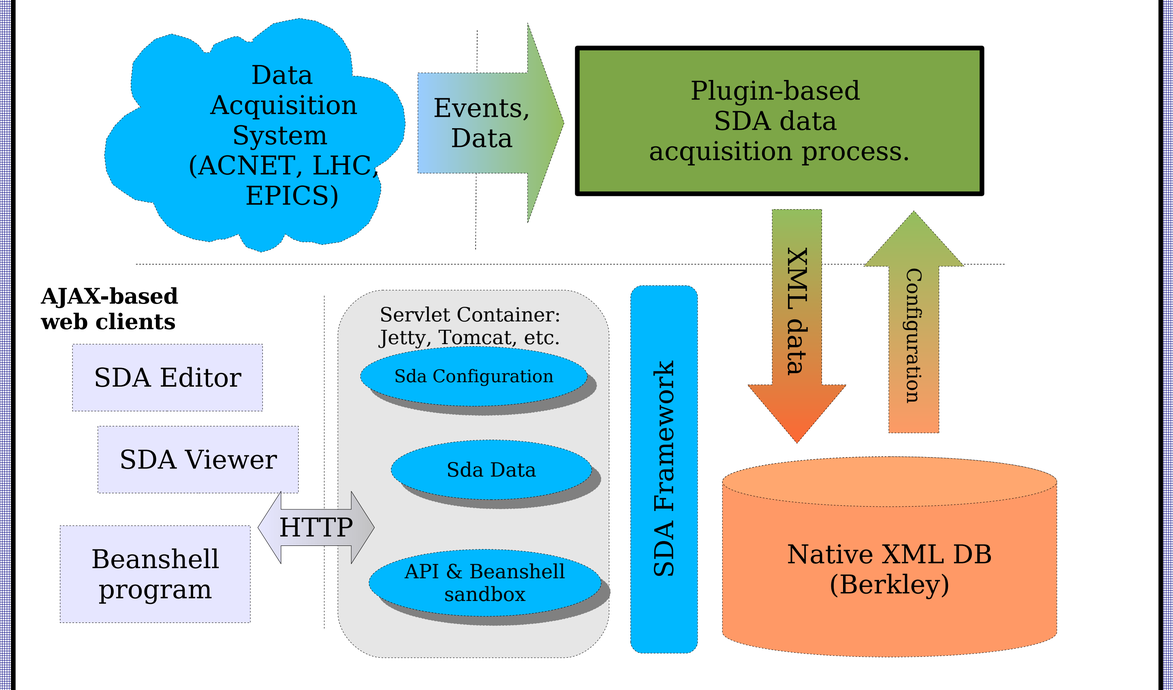


SDA background.

"SDA" means both "Sequenced Data Acquisition" and "Shot Data Analysis". It was developed at Fermilab for debugging and tuning the Tevatron accelerator chain.

The SDA system has been extremely useful during Collider Run II. Numerous applications were created to help scientists locate problems, performance bottlenecks, and study the accelerator physics issues during this run. Its main disadvantage is its deep integration into ACNET, which makes it impossible to use somewhere else.
Portable SDA solves this problem.

The overall diagram of SDA software.



SDA in these article means **Sequenced Data Acquisition**.

SDA is a logging system for describing and defining the beginning and the end of each step in a multistage process, in addition to defining steps within each stage. Each stage of this process defines different sets of properties and conditions that are collected.

SDA is **based on rules.** The significant terms of these rules are **event**, **device**, **collection** and **shot**.

Collection is a set of devices collected on specified events. Events for every device are described in the SDA configuration. A **shot** contains certain types of collections and rules for starting and stopping the processing of those collections. Every collection has a type and name associated with it, for example collection type 4 has the name "Inject Protons". Collections in one particular shot with the same type are called **Cases**. If a collection is repeated several times then the **Case** may have **Sets** - several instances of the same collection.

Shots, **Cases** and **Sets** are the main terms in SDA. They provide a common **time tick** understood by everyone.

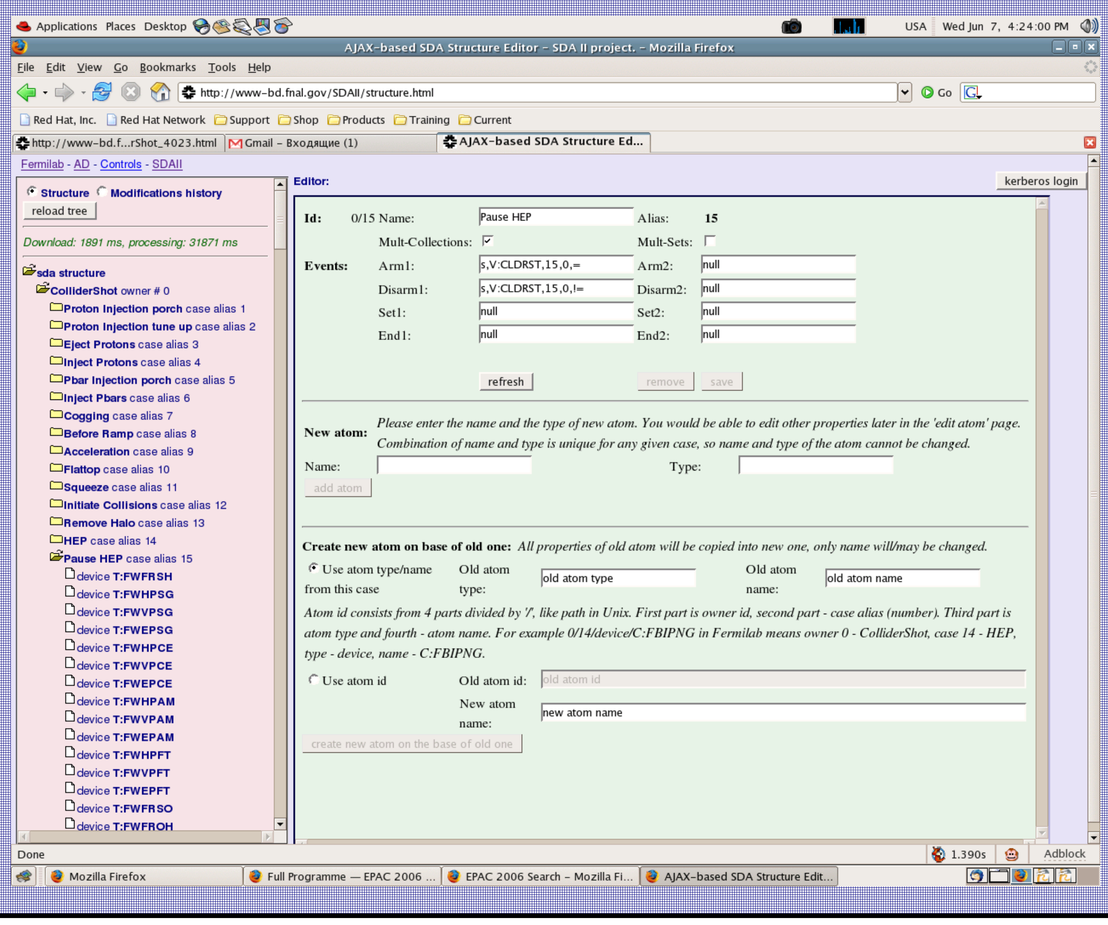
Shot, Cases, Sets, Devices describe a hierarchy and can be naturally represented as an XML document.

Berkley DB XML is used to store those documents. Berkley DB XML is an embedded XML database with XQuery-based access to documents stored in containers and indexed based on their content. Berkeley DB XML is built on top of **Berkeley DB** and inherits its rich features and attributes. Like Berkeley DB, Berkeley DB XML is a library, not a server, exposes a programmatic API for developers, and runs in process with the application.
More information: http://www.sleepycat.com/products/bdbxml.html

Experience: Usage of the native XML Database has simplified **significantly** the development of the system.

We consider the native XML Database as a significant advantage over a relational DB. Moreover, we can switch to a relational database if this is required because of modular structure of SDA.

Shot rules in SDA Editor.



All the server-side software for SDA is written in Java. The software is modular - plugins for different parts of the systems can be incorporated into the overall system.

The user interface is created with AJAX (Asynchronous Javascript and XML). This means that the SDA Viewer, the SDA Editor and the SDA Beanshell Sandbox are web pages. Web-startable Java applications can be incorporated easily into the system - some of the old SDA applications (Java-based SDA Viewer and OSDA library) can also interface to the new SDA system.

Java Analysis Studio (JAS) is planned to include into new SDA tools.