

State & Permit сервер для контроля за состоянием коллайдера HERA

R. Bacher

DESY, Hamburg, Germany

Ю. Нечаев, В. Соловьев

ГНЦ РФ Институт физики высоких энергий, Протвино, Россия

Описывается архитектура программного обеспечения State & Permit сервера для ускорительного комплекса HERA. State & Permit сервер состоит из 4 основных частей: DataServer, StateServer, PermitServer (Windows NT, Visual C++), работающих на среднем уровне СУ, и State & Permit Console (Windows NT, Visual Basic). State & Permit система реализована на базе клиент/сервер обмена данными с использованием Accelerator Component Oriented Programming (ACOP).

DataServer является интерфейсом между подсистемами HERA (уровень FECов) и "логическим ядром" State & Permit сервера. DataServer получает основной объем физических данных, параметров и статусных сигналов от подсистем HERA, преобразует их в логические (boolean) переменные и передает их в StateServer.

StateServer и PermitServer образуют "логическое ядро" системы, которое оперирует только логическими переменными для вычисления текущего, реального состояния коллайдера HERA и прогнозирования возможности его перехода к другим режимам работы.

State & Permit Console реализует интерфейс с пользователями системы, получает и отображает информацию о состоянии коллайдера, передает запросы оператора на изменение состояния ускорителей.

Полная система булевских уравнений, описывающая содержание "логического ядра", хранится в базе данных (Excel).

Введение

Коллайдер HERA является крупнейшим ускорителем DESY, он состоит из протонного кольца на основе сверхпроводящих магнитов на энергию 920 ГэВ и электронного/позитронного кольца на энергию 27.5 ГэВ. Настройка коллайдера HERA на рабочий режим является очень сложной последовательностью связанных процедур (этапов), начиная от тренировки (massage) сверхпроводящих магнитов и ввода протонного пучка до тонкой настройки коллайдерного взаимодействия пучков. При этом режимы работы магнитооптического оборудования многократно меняются при переходе от одной процедуры к другой. Управление коллайдером в этот момент ведет специальная программа (Sequencer), в работу которой в любой момент времени может вмешаться оператор. Однако Sequencer не контролирует корректность установки режимов работы оборудования, не проверяет всех условий возможности перехода на новый этап работы ускорителя. Эту задачу выполняет новый State & Permit сервер, который:

- Вычисляет текущее состояние коллайдера.
- Проверяет соответствие режимов работы подсистем коллайдера данному состоянию.
- Отображает и рассылает полную информацию всем заинтересованным пользователям.
- Дает разрешения (permits), пока как рекомендации, на переход коллайдера в новое состояние.

State & Permit сервер состоит из 4 основных частей: DataServer, StateServer, PermitServer (Windows NT, Visual C++), работающих на среднем уровне системы управления HERA, и State & Permit Console (Windows NT, Visual Basic).

StateServer и PermitServer образуют "логическое ядро" системы (Equation engine), которое оперирует только логическими переменными для вычисления текущего, реального состояния коллайдера HERA и прогнозирования возможности его перехода к другим режимам работы.

Обмен данными основан на TINE [1] протоколе, принятом в СУ HERA. State & Permit система реализована на базе клиент/сервер обмена данными с использованием Accelerator Component Oriented Programming (ACOP) [2].

1. Data Server

Основная задача DataServer (Visual C++) – получение полного объема физической информации с FEC'ов, управляющих подсистемами HERA, и преобразование данных в логические (Boolean) переменные, которые характеризуют состояние отдельных подсистем.

DataServer состоит из клиент и сервер частей. Как клиент DataServer получает каждую секунду необходимую информацию от 30 различных FEC'ов, объем получаемых данных различных типов (structures, integers, strings, floats, bytes) ~2000 единиц. Получение данных от FEC'ов синхронизовано за счет

использования возможностей ACOP (grouped links), что позволяет группировать запросы (сейчас их около 80) на различные FEC и вызывать программу обработки данных только после получения данных от всех процессов. Полную информацию об серверах, типах данных, их количестве и т.д. DataServer получает из локальной базы данных.

DataServer вычисляет около 70 выходных логических переменных для передачи в StateServer, каждая из которых может принимать 4 возможных значения (False = 0; True = 1; ошибка в вычислениях, например нет связи с сервером = 2; вычисление переменной пока не реализовано = 3).

DataServer получает информацию об изменении текущего состояния HERA от Sequencer'a и Permit & State Console, которая затем переформатируется и передается в StateServer, как логические intention flags (флаги намерений), их насчитывается ~150.

2. State Server

StateServer является Visual C++ приложением. Основная его задача – вычисление текущего состояния коллайдера HERA.

Вывод коллайдера на рабочий режим состоит из трех основных этапов:

1. Работа с протонным пучком (мода P).
2. Работа с электронным или позитронным пучком (мода E).
3. Настройка коллайдерного режима (мода E/P).

Таким образом, коллайдер HERA можно рассматривать как три различных ускорителя в зависимости от типа используемых в данный момент частиц. Соответственно база данных (и “логическое ядро”, описываемое этой базой данных) содержит три части – по одной на каждый этап работы коллайдера.

Ключевым элементом StateServer'a является класс “LogicElement”. Экземпляр этого класса можно рассматривать как обычный логический элемент (например, как в электронных схемах). Совокупность таких элементов образует логическую схему, описывающую содержимое “Equation engine”. Класс “LogicElement” имеет следующие свойства, описывающие его поведение:

Type – тип логического элемента: “AND”, “NOT AND”, “OR”, “NOT OR”, или “TRIGGER”. Этот набор типов предоставляет все необходимые возможности для создания конфигурируемой “Equation engine”. Это свойство определяется в базе данных для каждого экземпляра логического элемента.

ShouldBe – массив логических переменных, определяющих признак необходимости инвертирования входного сигнала. Считывается из файла базы данных.

Ignore – массив логических переменных, указывающих, учитывать или игнорировать данный входной сигнал при вычислении. Также считывается из файла базы данных.

Inputs – массив объектов, представляющих входные условия (название сигнала и его логическое значение) для каждого входа логического элемента. Эти входные условия поступают на входы логических элементов либо от DataServer'a, либо с выходов других логических элементов.

Output – логическая переменная, вычисленная по данным значениям входов с учетом типа элемента и значений ShouldBe и Ignore для каждого входа. Вычисление производится функцией Value() – членом класса “LogicElement”.

В момент запуска StateServer считывает соответствующий файл базы данных и создает в оперативной памяти полную логическую структуру “Логического ядра” в соответствии с ее описанием в этом файле.

Входные данные для StateServer'a поставляются DataServer'ом. Эти данные запрашиваются StateServer'ом в виде двух свойств:

- “STATE” (20 байт) содержит до 70 логических величин, описывающих состояние подсистем коллайдера в данный момент (например, “p collimators OK”);
- “STATE1” (40 байт) содержит до 150 логических intention flags, передающих команды к StateServer'у от Sequencer'a или от State&Permit Console (например, “e message ready”).

После получения входных данных от DataServer'a (периодичность опроса = 1 s) производится вычисление логической структуры, т.е. вычисление значений выходов всех элементов логической схемы.

Выходными данными StateServer'a являются значения тех логических элементов, которые характеризуют основное состояние (например, “p MESSAGE”), подсостояние (например, “completed”) и атрибут (VALID или NOT VALID). Эти данные, а также данные о режиме работы ускорителя (P, E или E/P) и заряде электрон/позитронного пучка, упакованные в 1 long, формируют свойство “CONDATA”, доступное всем заинтересованным клиентам, в том числе PermitServer'у.

Кроме того, в процессе вычисления логической структуры производится анализ причин, по которым какое-то конкретное состояние имеет атрибут "NOT VALID" (например, если сигнал "p magnets OK" имеет значение FALSE, то состояние "p MESSAGE ongoing" будет иметь атрибут "NOT VALID"). Информация об этих т.н. "wrong bits" содержится в свойстве "REASONS" (18 longs) и используется в State & Permit Console – привилегированный оператор может потребовать игнорировать данный сигнал при вычислении логической структуры.

Логическая структура StateServer'a может быть перезагружена привилегированным оператором посредством запроса от State & Permit Console без перезапуска StateServer'a.

3. Permit Server

Основная задача PermitServer'a – вычислить и передать пользователю информацию о том, какие изменения в состоянии коллайдера HERA возможны в данный момент времени.

PermitServer во многом похож на StateServer, однако он использует свою независимую базу данных для 'Equation engine' и использует данные, получаемые от StateServer'a и DataServer'a. Соответственно PermitServer вычисляет 64 выходные логические переменные.

Логическая структура PermitServer'a может быть перезагружена по запросу пользователя с Permit & State Console без перезагрузки PermitServer'a.

4. STATE & PERMIT Console

State & Permit Console (Windows NT, Visual Basic) является графическим интерфейсом между оператором/пользователем и Data/State/Permit серверами. На консоли отображаются текущие состояния коллайдера, все возможные разрешения на переход в другие состояния, общие физические параметры пучков, время последних переходом и т.д. Кроме того, на консоли показываются причины (если они реально существуют в данный момент), из-за которых текущее состояние коллайдера диагностируется как не возможное.

Существуют дополнительные возможности для ограниченной группы пользователей по установке начальных состояний коллайдера, его перевода в произвольное состояние и т.д.

Все пользователи разбиты на три категории:

- "Master user" – единственный пользователь системы, имеющий полные права на изменение состояния и подсостояний коллайдера, перезагрузку баз данных, включение/выключение записи в logfile и т.д.
- "Expert user" – пользователь, имеющий права "Common user", но имеющий возможность стать "Master user", если такового в данный момент времени нет в сети.
- "Common user" – пользователь, имеющий возможность только получать информацию и пассивно следить за процессом.

Вся передача информации от/к State & Permit Console идет с использованием ACOP.

Литература

[1] P. Duval. "TINE: An Integrated Control System for HERA", PCaPAC '99.

[2] I. Deloose, P. Duval, H. Wu. "The Use of ACOP Tools in Writing Control System Software", CERN/PS 97-67 (CO), 1997, Int. Conference on Accelerator and Large Experimental Physics Control Systems (Icalepcs'97), November 3-7, 1997, IHEP, Beijing, China.