

# Пакет YAQ как унифицированный программный интерфейс в системах управления ускорительными комплексами

Е.А.Жарков, В.И.Ковальцов  
ГНЦ РФ Институт физики высоких энергий, Протвино, Россия  
Ж. Люис  
ЦЕРН, Женева, Швейцария

## Введение

Системы управления ускорительными комплексами (СКУ) включают множество программ, выполняющихся в распределенной неоднородной среде и взаимодействующих между собой в процессах контроля оборудованием ускорителей, сбора данных и их обработки и визуализации. Поэтому важной является проблема создания механизмов взаимодействия программ, обеспечивающих как высокую эффективность, так и гибкость, простоту и надежность [1, 2]. В данной работе обсуждается программный пакет YAQ, предназначенный для унификации межпрограммного интерфейса в СКУ.

## 1. Основные понятия

К основным понятиям пакета относятся такие термины, как устройство, семейство устройств, методы устройств. Устройство — это элемент оборудования или программный объект. Доступ к устройству осуществляется через методы или акции. Устройство принадлежит к некоторому семейству или классу. Все устройства данного семейства имеют один и тот же набор акций. Здесь отсутствуют понятия подкласса и суперкласса, а следовательно, не реализуется и концепция наследования методов и нет иерархии классов, что важно в объектно-ориентированном программировании.

## 2. Схема взаимодействия программ

Пакет YAQ поддерживает традиционную схему взаимодействия клиент–сервер [3], при этом программа–сервер (или сервер устройств, СУ) сама может быть клиентом другого сервера устройств. По отношению к устройству программа–клиент (или прикладная программа, ПП) может запросить выполнение тех или иных методов данного устройства; она не видит его внутренней организации. В то же время сервер устройств является хранилищем объектов, в нем содержится реализация всех методов устройств данного семейства. Он также создает и уничтожает внутреннее представление устройства — его данные.

При обращении к устройству ПП указывает его имя (оно должно быть уникальным в СКУ), затем имя вызываемого метода и список параметров, необходимый

для данного метода данного устройства. Таким образом, обобщенный вызов метода устройства имеет форму `uqCall (device, method, arguments)`, где первый аргумент является уникальным идентификатором устройства, второй задает имя метода, а следующие служат его аргументами и могут отсутствовать.

Здесь, как и в ООП [4], имеет место полиморфизм операций (методов), т.е. одной и той же операции соответствуют разные действия, когда она применяется к объектам разных классов. Только при первом обращении к устройству происходит неявное обращение в центральную базу данных реального времени (ЦБДРВ) с тем, чтобы определить, какой именно сервер устройств содержит заданное устройство и каковы методы его семейства. Фактические параметры метода должны соответствовать его формальному описанию и при вызове не контролируются; при формировании запроса к СУ все данные автоматически преобразуются в сетевой формат, а результат выполнения метода преобразуется в локальный (машинный) формат.

Работа сервера устройств заключается в декодировании поступающих запросов, обратном преобразовании параметров метода в локальный формат, идентификации устройства и метода и вызове соответствующей процедуры. Результат выполнения процедуры кодируется в сетевое представление и сервер передает ответное сообщение программе-клиенту.

В качестве сетевого протокола выбран UDP [3], обеспечивающий высокую эффективность, неограниченное количество клиентов для любого сервера, возможность использовать широковещательную передачу пакетов.

### 3. Организация базы данных описаний

Центральная база данных реального времени представляет собой `dbm`-файл, дающий возможность чтения и записи в нее по единственному ключу. Эта БД может динамически расширяться, если в СКУ появляются новые компоненты. Все данные в этой БД представлены в сетевом формате, поскольку сеть ЭВМ в СКУ в общем случае неоднородна. БД хранит описания серверов устройств, собственно устройств, семейств и их методов, причем наряду с обязательной частью, общей для всех приложений YAQ, эти описания содержат формат специфичных для того или иного приложения данных, которые также обязаны передаваться при вызове методов устройств. Например, в некоторых случаях может существовать код устройства или код метода (и быть разным по длине и даже типу), необходимый для идентификации устройства или метода. Если специфичная часть не пуста, то ее данные также присутствуют в БД и выбираются при первом обращении к устройству. Наличие такого рода специфики в каждом описании дает определенную свободу проектировщику семейства устройств, каким образом именно идентифицировать устройства, семейства, метода для программы-сервера.

Доступ к ЦБДРУ реализуется стандартным для пакета YAQ способом, т.е. посредством вызова методов предопределенного устройства — сервера базы данных. Таким образом гарантируется инкапсуляция внутренней структуры данных в БД. В общем случае можно сказать, что серверы устройств инкапсулируют данные своих устройств, гарантируя внешний интерфейс к устройствам и оставляя свободу их внутренней реализации за разработчиком.

## Заключение

Функционально пакет YAQ объединяет как возможности RPC [3], обеспечивающего прозрачный вызов удаленных процедур, так и возможности объектно-ориентированного программирования с его унифицированным и гибким механизмом передачи сообщений объектам [4].

Пакет YAQ используется в СКУ комплексом ускорителей CERN PS на его основе построена система управления распределенными таблицами DTMRT [5], широко применяемая в таймерной системе и программном интерфейсе к ней со стороны рабочих станций и DSC. Данный пакет также входит в ПО системы управления каналом перевода пучка из бустера ИФВЭ в У70, построенной на основе вертикального среза: рабочая станция – VME-крейт – крейт MB-I – крейт СУММА.

## Список литературы

- [1] Franck Di Maio, Alessandro Risso. The CERN-PS Equipment Access Library. Software specifications. Version 3. PS/CO/Note 93-87 (Spec.). CERN, February 1994.
- [2] Pierre Charrue. Accessing Equipment in the SPS-LEP Controls Infrastructure: "The SL-EQUIP package". Software User Manual. SL/Note, 93-86 (CO). CERN, September 1993.
- [3] P.Anderson, V.Frammery, G.Morpurgo. User Guide to the Network Compiler Remote Procedure Call (NC/RPC). LEP Controls note 97. CERN, May 1989.
- [4] Brad J. Cox. Object-Oriented Programming: An Evolutionary Approach, Addison-Wesley Publishing Company, 1986.
- [5] V.Kovaltsov, J.Lewis: Distributed memory in a heterogeneous network, as used in the CERN PS complex timing system. ICALEPCS, Chicago, U.S.A. Oct.29-Nov.3, 1995.