VLT Software Workshop - May 97

CCS OO Development Tools:

Data Base Loader - ECCS - EVH

G.Chiozzi

14/05/97

Contents

- Introduction
- The DBL Database Loader
- Extended CCS
- Event Handling Toolkit
- Conclusion

Introduction

The VLT Software makes extensive usage of Object Oriented design and development methodologies.

In order to support the Object Oriented approach, CCS provides 3 cornerstone components:

- Online-Database Loader
- Extended CCS
- The Event Handling Toolkit

They are standard components of CCS available since end of 1994 and since July 1995 release they are very stable.

A lot of care has been spent in the initial design, so that no big changes and only limited extensions have been necessary after that date.

The DBL database loader

All data that can be of interest for external modules, to get a picture of the status of the system or for system's tuning and configuration are stored in the real-time database

The database is a corner-stone of the VLT software architecture and must be uniform and coherent through all the applications developed internally and by consortia, based or not on OO concepts.

- It is a real-time distributed database, based over the hierarchical model
- We have developed a database loader and a class browser that extend the semantic and syntax of the RTAP database definition language introducing OO concepts like:
    - inheritance
    - overloading
    - methods, implemented with C++ classes
    - a "database class" is a new structured data type

DBL: major changes since JUL95

NOV96 contains a set of new features.
Previous versions contain mainly bug fixing and very few extensions.

Changes introduced with NOV96 release:

- Extension of syntax to allow definition of vectors of points.

```
 LOOP 50 ATTRIBUTE SOME_CLASS point_#

would be equivalent to this:

      ATTRIBUTE SOME_CLASS point_0
      ATTRIBUTE SOME_CLASS point_1
        ...
      ATTRIBUTE SOME_CLASS point_48
      ATTRIBUTE SOME_CLASS point_49
```

- Default values for vector and table entries.

```
ATTRIBUTE TABLE myTable (20,
                         rtINT16    field1  99,
                         rtINT16    field2,
                         rtINT16    field3  4,
                         rtLOGICAL  field4
                         )
ATTRIBUTE VECTOR myVector (4, rtINT16 3)
```

- Initialization of tables and vectors.

```
 ATTRIBUTE TABLE table1 (10,
                           rtINT16 f1,
                           rtINT16 f2,
                           rtINT16 f3,
                           rtINT16 f4)
  BEGIN
       Value (5, f3) ((10, 20), (30, 40))
       //     ^   ^
       //     |   Starting field name
       //     Starting record number
  END

  ATTRIBUTE VECTOR vector1 (15,rtINT16)
  BEGIN
       Value (8) (123, 456, 789)
       //  ^
       //  Starting record number
  END
```

- Enhancing overloading of tables to allow addition of new fields during overload.

```
CLASS BASE_CLASS BASE_TABLE
BEGIN
     ATTRIBUTE TABLE someTable (0,
                                rtINT32 fieldA,
                                rtINT32 fieldB
                               )
END
CLASS BASE_TABLE DERIVED_TABLE
BEGIN
    ATTRIBUTE TABLE someTable (0, ...,
                            rtINT32 fieldC,
                            rtINT32 fieldD
                         )
END
```

Extended CCS

Extended CCS is a library written in C++ on top of CCS. It contains functions and simple classes and requires a minimal knowledge of C++ syntax.

Purpose of the library is:

- Provide easier usage of common CCS features
- Embed in functions or simple classes commonly used application patterns
- Provide base CCS classes for the development of truly object oriented applications
- Prevent common errors in applications
- Standardise produced code

Only the commonly used parts of CCS are integrated in ECCS.

ECCS: major changes since JUL95

DEC95

- Added trace feature for received and sent messages/events
- Added classes for access to database TABLES and VECTORS

JUN96

- Added support for CCS alarms
- eccsTestDriver: Added support for filters

NOV96

- This release contains mainly bug fixing. No new major features have been introduced.

MAY97

- Shared libraries

Event Handling Toolkit

The Event Handling Toolkit is a set of C++ classes to develop event driven applications on workstations.
Development of applications requires a knowledge of concepts for event driven programs and object-oriented programming techniques.

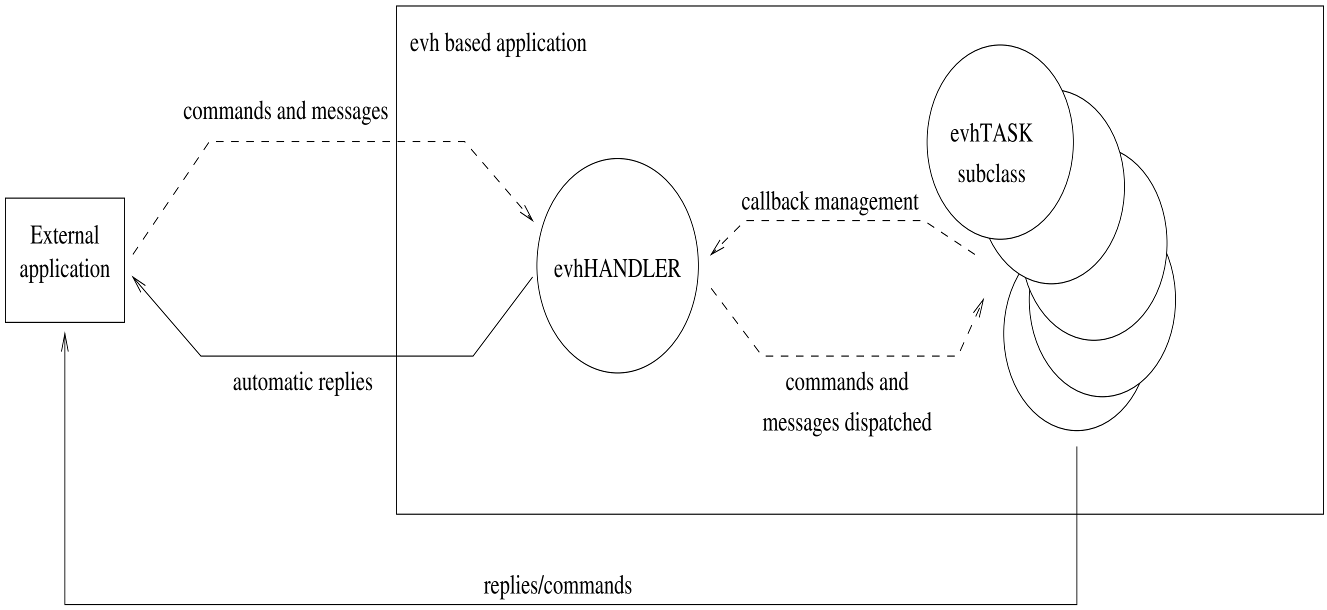
It runs only on Workstation.

Purpose of EVH is:

- Provide a framework to implement easily workstation applications that receive and send commands and messages.
- Implement in a standard way common application patterns
- Prevent common errors in applications
- Support object-oriented design and development
- Provide tools for testing and debugging of applications.

The EVH main loop

- Processes are designed in an event driven way
- The implementation is based around the *evhHANDLER* class



The design of every process consists mainly in the design of the independent objects providing the methods to be attached as callbacks to the events

EVH: major changes since JUL95

Most important extensions are new diagnostic features

DEC95

- evhCRONOMETER a simple cronometer
- evhHANDLER:
  Added new features for tracing received/sent messages.
  Added new features for monitoring of callback execution time.
  Added new convenience functions to handle callback lists:

JUN96

- The MAIN CHANGE in this release of evh library consist in a modification in the way database events are handled.
- Added handling of alarm events
- evhHANDLER.C:
  Check on error stack for pending errors

NOV96

- evhSIMPLE_TASK:
  Added method to turn ON/OFF warning on Exit() with multiple objects. By default is ON.
- evhCALLBACK_DICT:
  Added memory collector for deleted callbacks. This makes more robust removing callbacks from within a callback.

MAY97

- Shared libraries

Conclusion

This presentation has shown only some of the most important aspects of these development tools.

For more details look at:

- VLT Software - CCS On-Line Database Loader User Manual (VLT-MAN-ESO-17210-0707)
- VLT Software - Extended CCS User Manual (VLT-MAN-ESO-17210-0770)
- VLT Software - CCS Event Tool Kit User Manual (VLT-MAN-ESO-17210-0771)
- Code examples in <mod>/test directories.