

**ConSys - A new  
Highly Object  
Oriented Control  
System.**

**J.S. Nielsen, K.T. Nielsen,  
and T. Worm**

**ISA, University of Århus,  
Denmark**

# Introduction

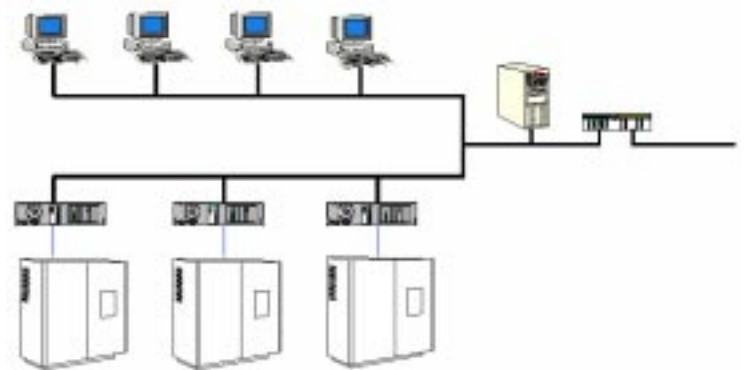
A new control system, ConSys, for the accelerators at ISA, Århus has been developed. The control system follows the standard

model, as well as the publisher/subscriber model. The resulting system is a highly modular control system based on object



## Features

- Object Oriented
- Symmetric - same core code on front-end and console computers
- Windows NT only.
- Publisher/Subscriber model
- Easy to maintain: Simple interface objects to client applications and devices.
- All configurations including parameter definitions are stored in a central database.
- General applications are machine independent.
- Dynamic load of device drivers and data servers.



# Abstract Model

The ConSys control system is designed using an object oriented approach. A very limited number of base objects implement the core functionality:

## The Device object:

Provides the abstraction between the hardware instrument drivers and the control system. Devices contain the real time database needed to store the parameters for a given device. The device object includes a few abstract methods for core functionality, like reading, writing and registration at the device. The device signals registered [Data Servers](#) when data has changed.

## The Data Server object:

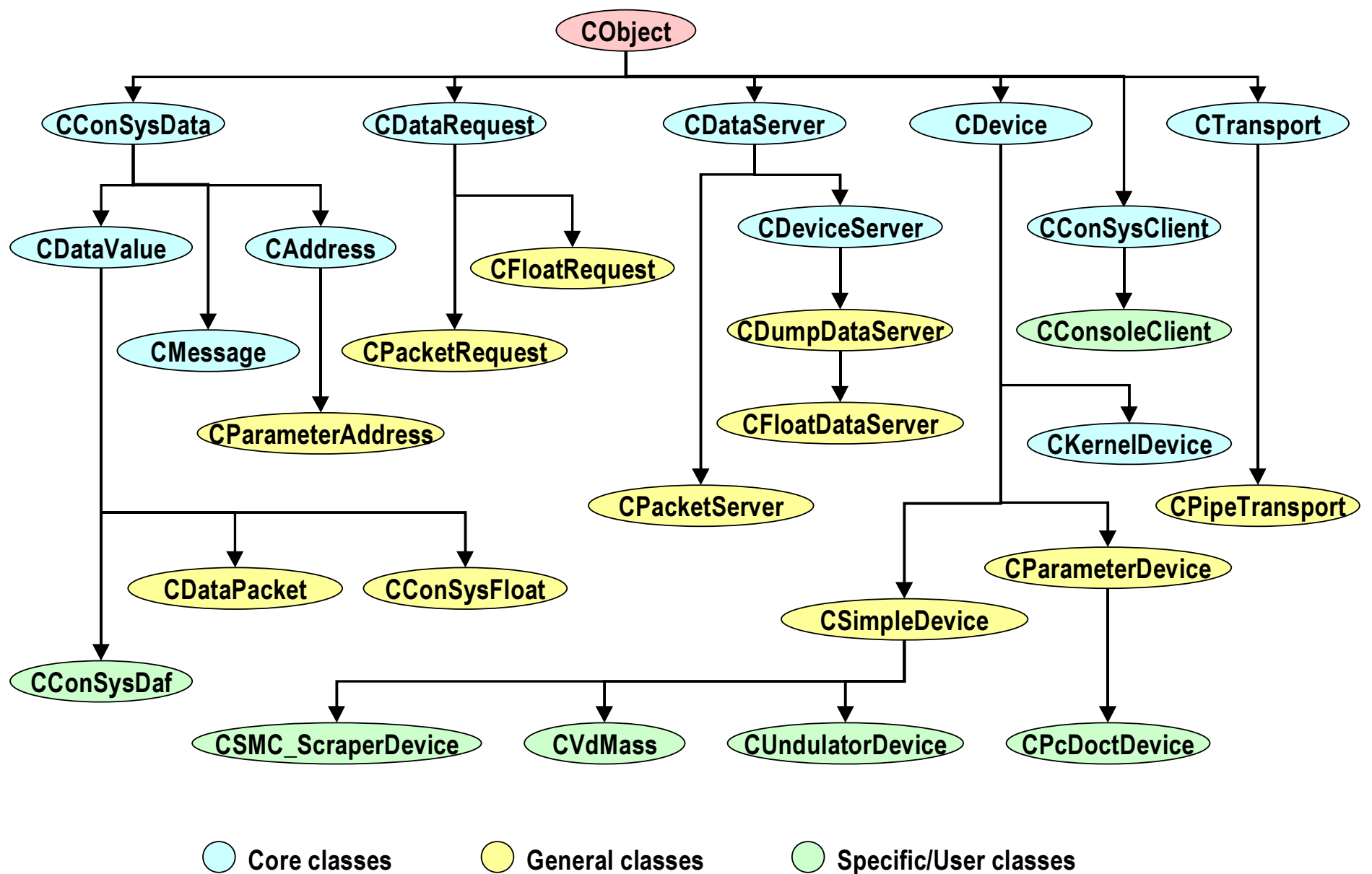
Obtaining and transmitting data to the client is done by data servers. A data server may register itself at a device. The device will signal the data server when data may have changed. It is the responsibility of the data server to decide whether the data should be send to the client or not - based on criteria set from the client application.

## The Data Request object:

The data request contains the information needed by the control system to load and initialise a data server and connect it to a device.

## The Client object:

Applications interface to the control system through the ConSys client object. This object provides a small but powerful set of methods to be used by the client program to gain access to the control system data. On creation the client is supplied with a data request. Based on the data request the client will create all necessary connections.



## The ConSys Data Object:

The base of all data transmitted in the control system is the ConSys data object. There are three important descendants: The **Message object**: The base for all messages transmitted in the control system.

The **Data Value** object: Base class for all data values transmitted in the system.

The **address** object - see below.

## Addressing - the address object:

- **Generalised address space.**

The address object hierarchy defines a simple, uniform and yet flexible way of addressing parameters.

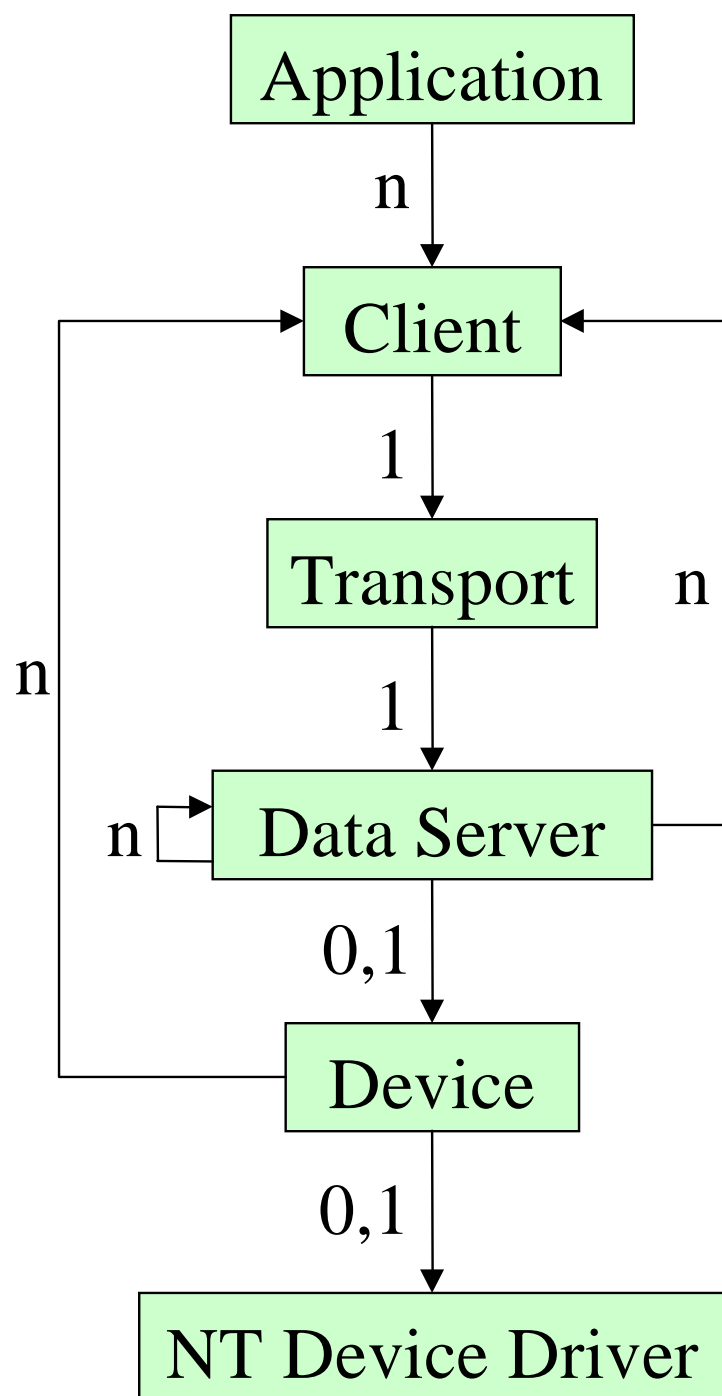
- **Simple base address object.**

Only contains information on how to locate the device. This is all the control system needs to access the device. Descendant address objects contain additional information needed by the device to locate a parameter.

- **Fully qualified address object.**

The address object includes all information needed to locate a parameter.

# Object Relations



The figure to the left shows the relations between the most important objects in the control system.

An **Application** may create as many **clients** as it needs to gain access to the control system.

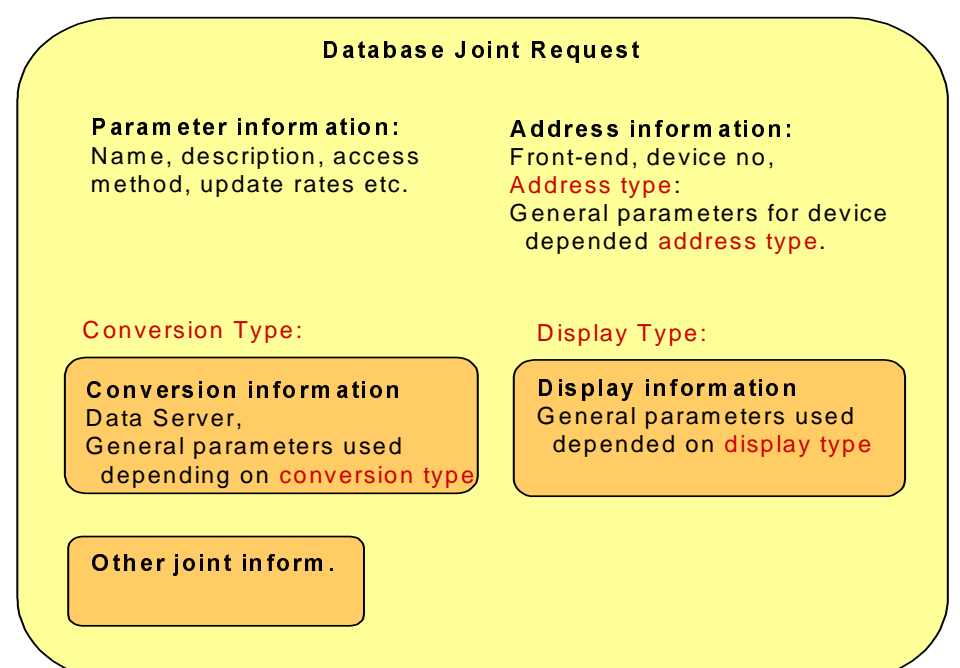
Each **client** creates a **transport**, which in turns creates a single **data server**.

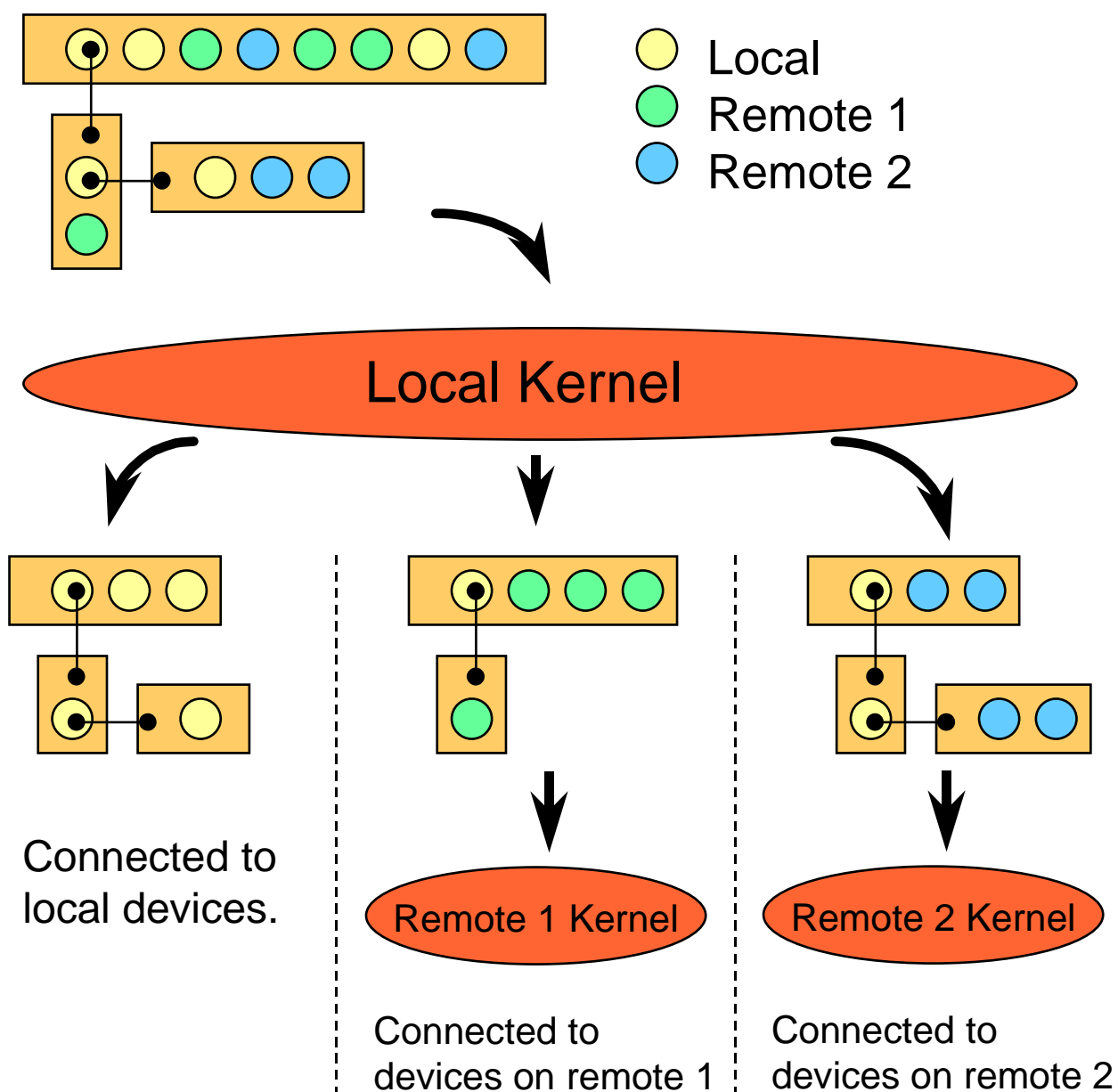
The specific type of the data server depends on the request specification. The data server

may create as many helper data servers or new client objects as it needs. It may also connect to a single **device**. A device may create as many client objects it wants. If the device is a virtual device it will not connect to a device driver. If the device is controlling hardware, it connects to a single **device driver**.

## Database

- Used to store all machine/hardware dependent information - including information needed for client programs to generate displays.
- Any ODBC database can be used. We use the Microsoft SQL server.
- Initialisation data for ConSys objects stored in generalised, flat tables





## Establishing a data connection

- Application: Construct a data request.**  
 The basic request includes the address object needed to locate the parameter, and specifies the dataserver object to be used with the request.
- Application: Create client.**  
 The application creates a client and initialises it with the data request.
- Client: Create transport**  
 If the request is for a single front-end, the client creates a transport object and sends the request to the transport.
- If the request is a packet request containing requests for multiple front-ends, the packet request is split into multiple packet requests, each containing a request for a single front-end.
- Transport: Create data server**  
 The transport now connects to the ConSys kernel on the computer specified by the request. The receiving side of the transport load and create the data server specified by the request.
- Data Server: Register at device.**  
 If a data server is a single parameter data server it registers itself on the device addressed by the request, and establishes a connection to the transport layer for communication of data between itself and the client.
- Packet data server.**  
 A special dataserver, the packet server, handles the packet request. It is the packet data server that is responsible for the machine dependent sorting and remote connection mentioned above.

# Experiences/Conclusion

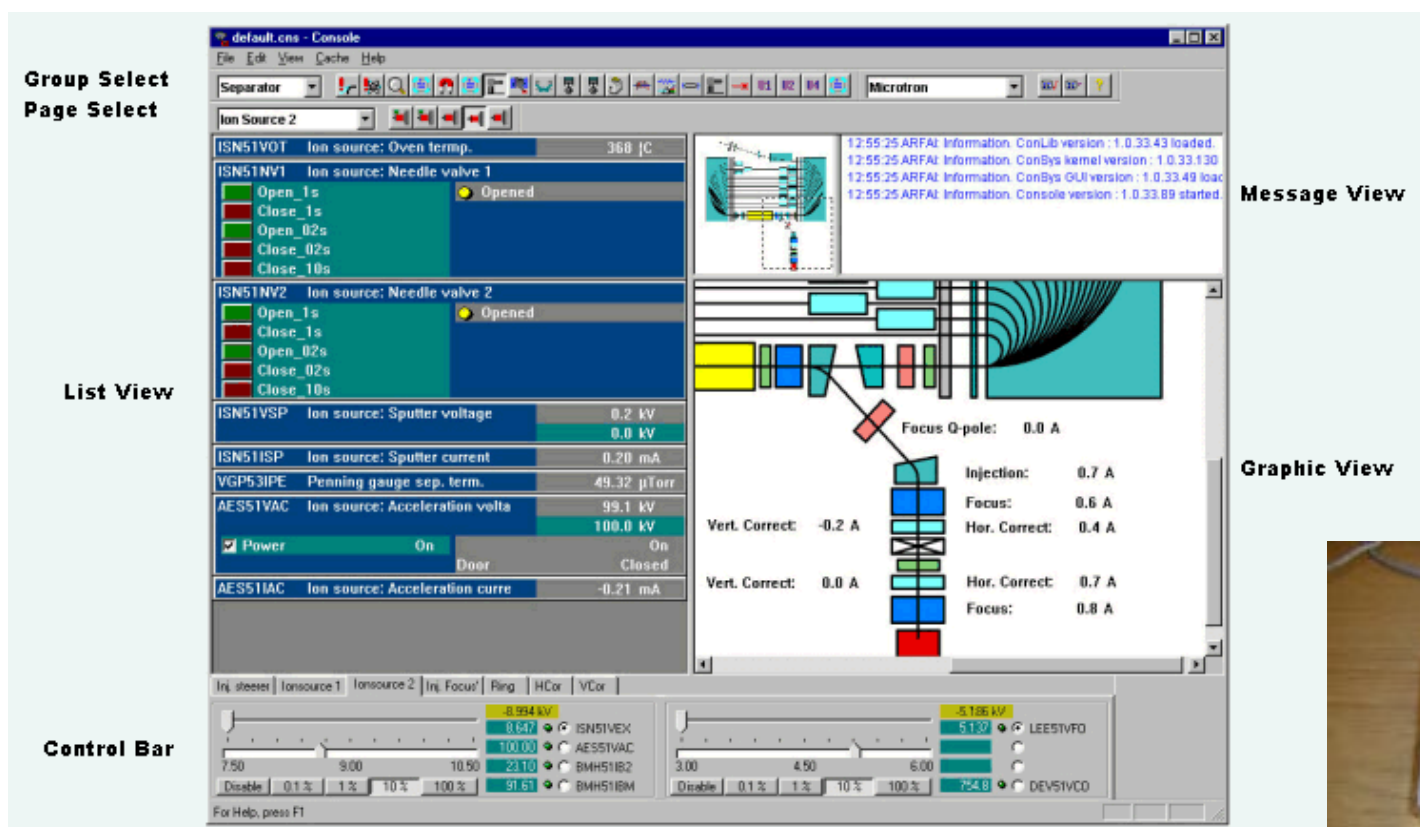
- **Simple design**  
The object-oriented approach has resulted in a highly flexible and modular design, where front-end and client computers have identical core software. A very limited number of base objects defines the core of the system.
- **Performance**  
Provides real time (analogue) feeling for operators - typical update rates are 5 Hz.
- **Stability**
  - **Windows NT (servers, front-ends), SQL server, network:** Very good - almost no failures.
  - **ConSys: Reasonable;**
    - **Front-end with PCDoct:** Good - almost never down.
    - **Console PC's:** Applications may fail - but the kernel on the PC survives in most cases.
- **Limitations**
  - **Strong ties to Windows NT & MFC.**
  - **At the moment no interface to other platforms.**
  - **Not drivers for all hardware (especially old hardware)**

# More information

- <http://www.isa.au.dk/ConSys>  
This is the homepage of the the ConSys system. Here you will find all the online documentation.
- <http://www.isa.au.dk/ConSys/pdf/ICALEPCS99Poster.pdf>  
This poster.
- <http://www.isa.au.dk/ConSys/pdf/ICALEPCSArticle.pdf>  
The accompanying article.
- <http://www.isa.au.dk/ConSys/pdf/EPAC98ConsysPoster.pdf>  
EPAC98 ConSys poster.
- [http://www.isa.au.dk/ConSys/pdf/EPAC98-tup46g\(ConSys\).pdf](http://www.isa.au.dk/ConSys/pdf/EPAC98-tup46g(ConSys).pdf)  
EPAC98 ConSys article - overview.



# Examples of client programs



## Console

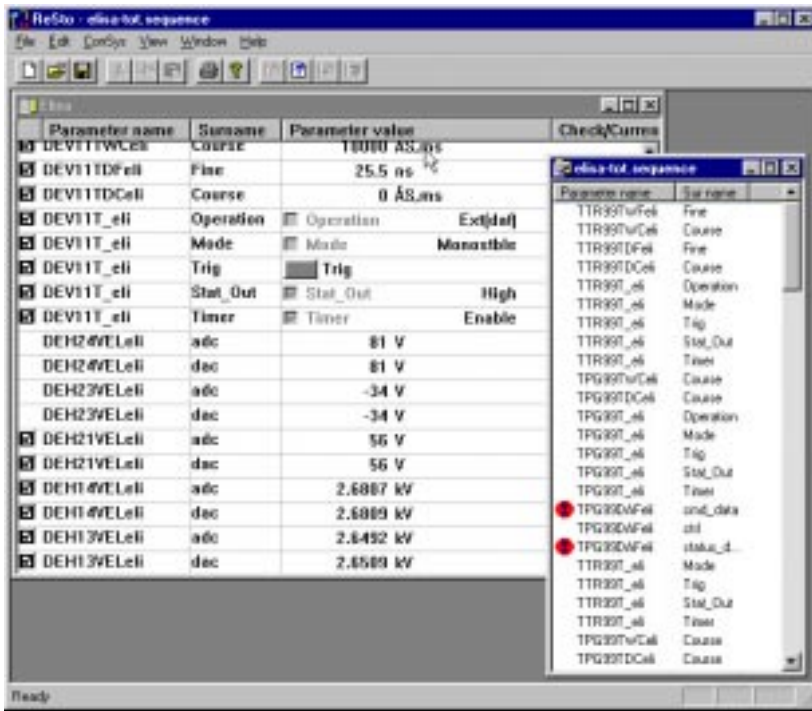
Main program to display and adjust parameters.

- Machine independent - all console pages are fully defined in the SQL database.
- Two control bars with 'analogue' display of two parameters. Each one selectable from a list of 4 parameters.
- 'Analogue feel' through digital potentiometers.
- Mixed graphic and text based selections.
- Drag & Drop of parameters to control bars or other Con-Sys applications.

## Digipot

To achieve an 'analogue feel' when controlling the parameters, each console computer has a box with two digital potentiometers, which can be used to adjust the two active parameters in the Console control bar.

- Works by sending messages to the active window.
- Connects directly to a standard serial port.
- Can be installed as a service.
- With buttons for push and pop parameter settings, and for selection of active control parameters.



# ReSto

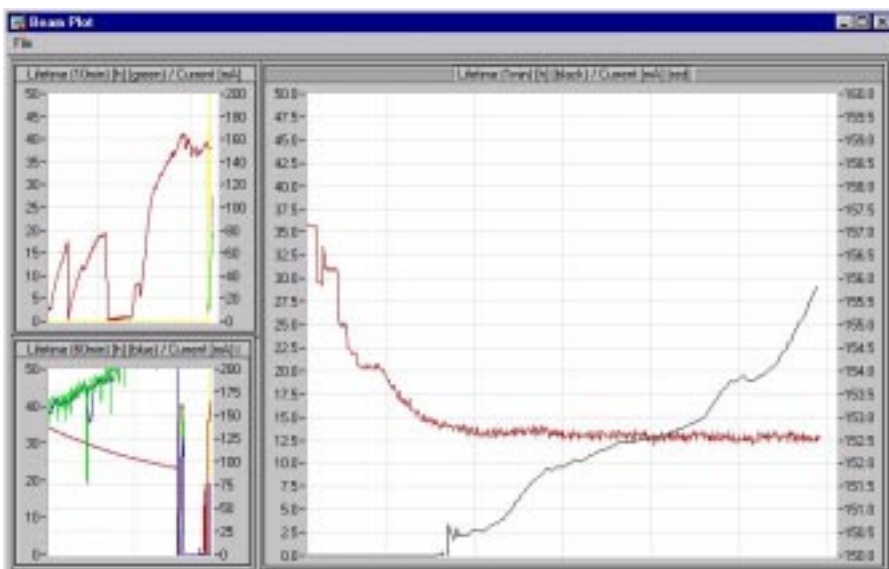
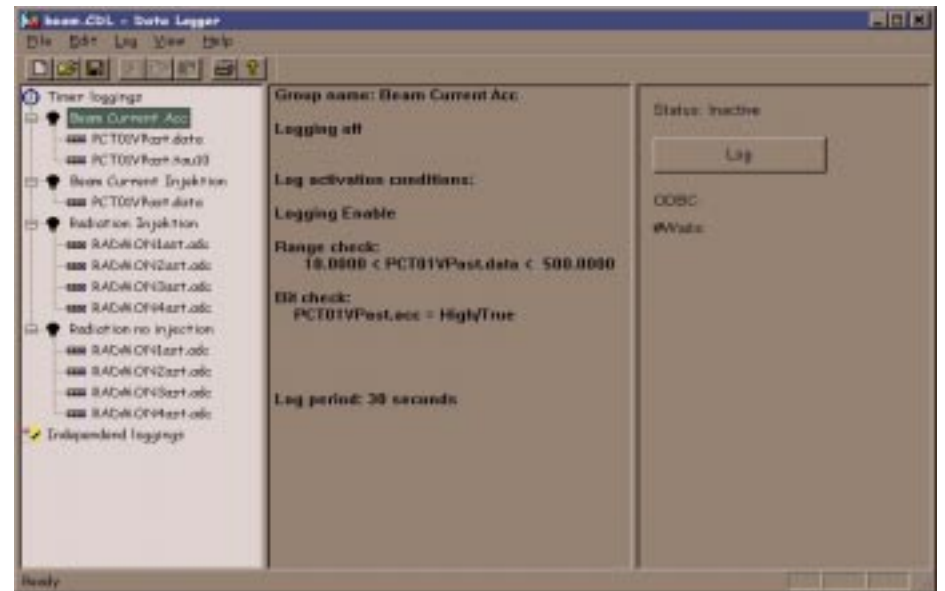
A general purpose program to store and restore parameters on the system.

- Upload parameters from one or more machines (or machine parts).
- Download parameters specified in a sequence file.

# Datalogger

A general purpose program to log run-time data values to the SQL database.

- Logging conditions (binary and floating point ranges).
- Rates up to 1 Hz.



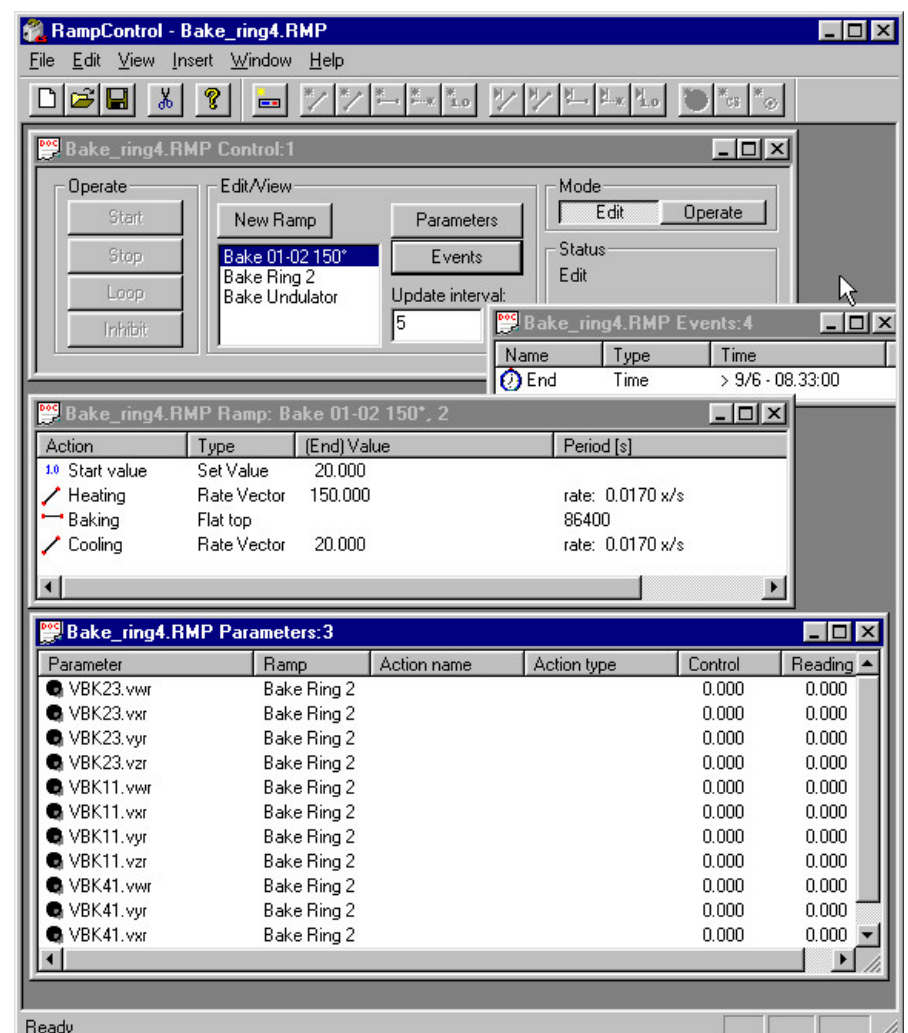
# BeamPlot

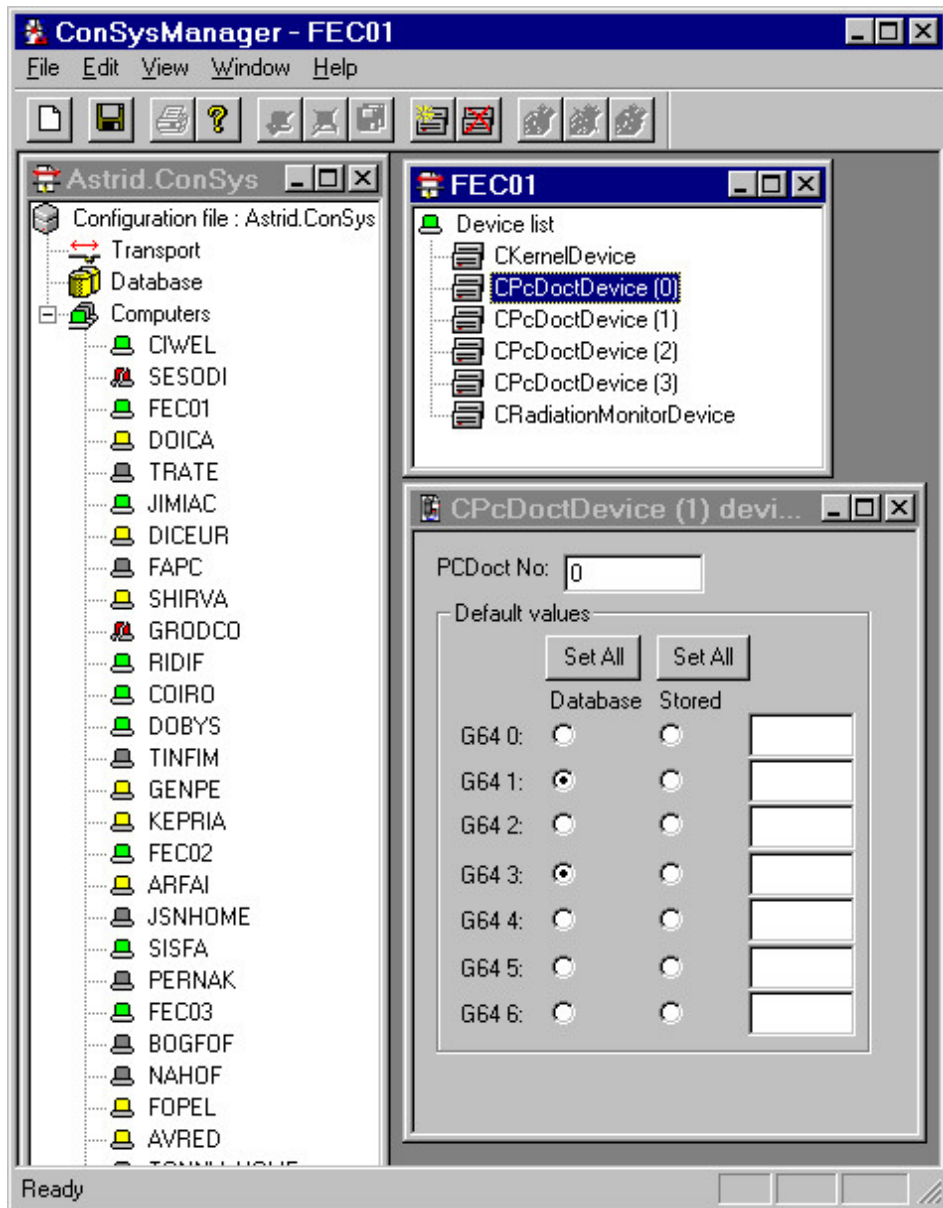
A specific program to plot beam current and lifetimes during electron runs.

# RampControl

A general purpose program to perform (slow) ramping of parameters.

- Ramp floating point values in either specified time or at specified rate.
- Can also set binary values.
- Can wait for specified events.
- Rates up to 1 Hz.





# ConSysManager

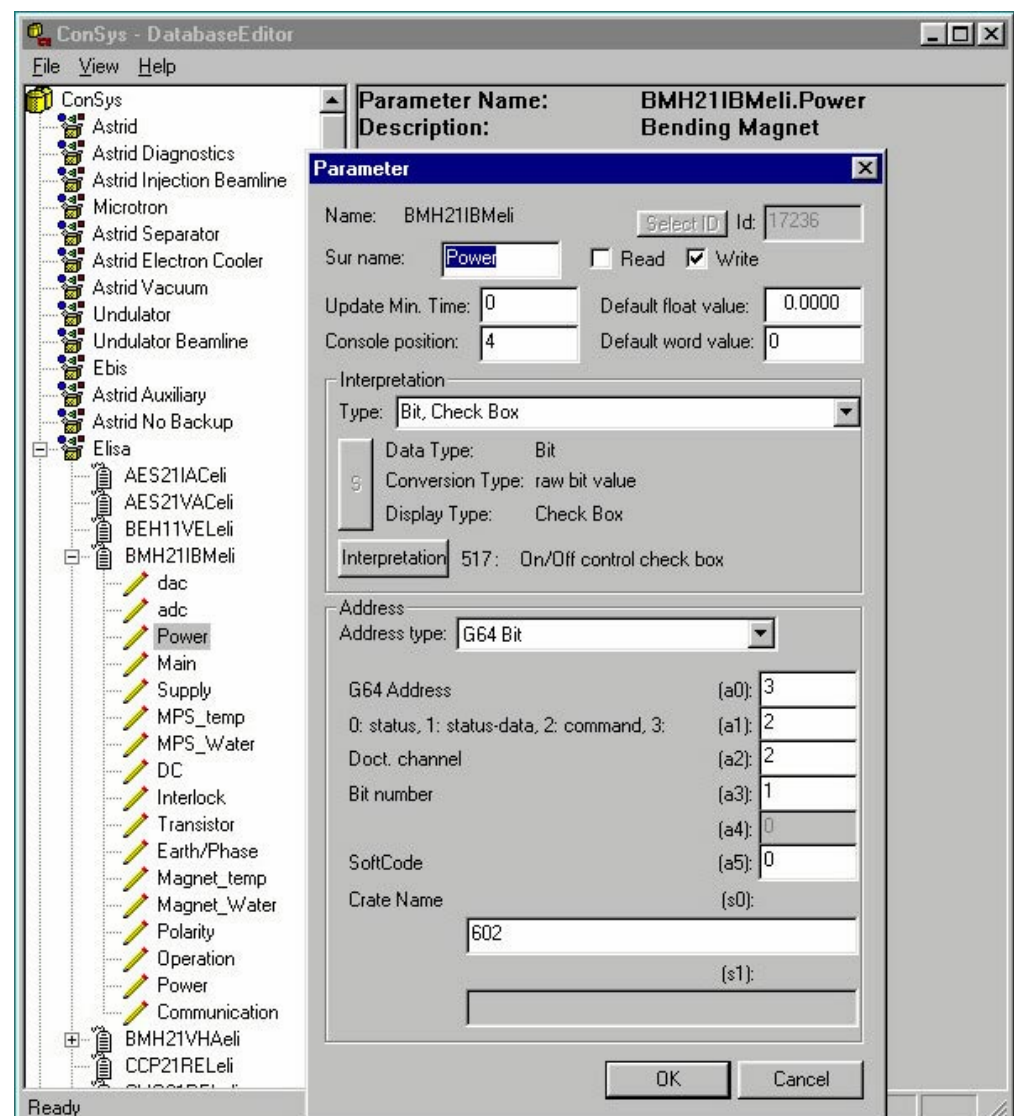
The system setup program.

- Specification of system database and communication protocol.
- Specification of which computers participate in the system.
- Specification of which devices to be loaded on participating computers.
- Setup of devices on the participating computers.

# Database Editor

A system program to ease setup and maintenance of the system database.

- Object oriented editing of the ConSys tables - based on description tables.
- Ease maintenance and secure data integrity.
- Used together with Access for editing ConSys SQL data.



# Commercial Tools

**Policy: Commercial tools used whenever possible:**

**+: Fast development cycle, professional look.**

**- : Limitations (bound to one solution).**

- **Microsoft Visual C++ 6.0**

The primary development tool. The control system has a strong tie to MFC, Microsoft Foundation Classes. The communication protocol is based on pipes, an intrinsic part of Windows NT.

- **Microsoft SQL server**

Used to store all configuration data.

- **Microsoft Excel**

Used as an editor for the DAF-files (function tables).

- **Component Works (National Instruments)**

ActiveX controls, used for plotting and technical controls.

- **CVI (National Instruments)**

Can be compiled into the ConSys code. Used for device drivers and some displays. Also used as mathematical library.

- **WWW Browser**

Used for documentation, and for status display. In the future possibly also for control (Java).

- **Visual Basic, Excel, Luprise Delphi, LabView etc.**

Is supported through the ConSysAPI (a Windows DLL) and the ACOP ActiveX interface for user programs .