TANGO INTEGRATION OF A SIMATIC WINCC OPEN ARCHITECTURE SCADA SYSTEM AT ANKA

T. Spangenberg*, K. Cerff, W. Mexner, KIT, Karlsruhe, Germany
Volker Kaiser, Softwareschneiderei GmbH, Karlsruhe, Germany

Abstract

The WinCC OA [1] supervisory control and data acquisition (SCADA) system (previous PVSS II) provides at the ANKA synchrotron facility a powerful and very scalable tool to manage the enormous variety of technical equipment relevant for house keeping, beamline, and machine operation. Crucial to the applicability of a SCADA system for the ANKA synchrotron are the provided options to integrate it into other control concepts even if they are working e.g. on different time scales, managing concepts, and control standards.

Especially these latter aspects result into different approaches for controlling concepts for technical services, storage ring, and beamlines.

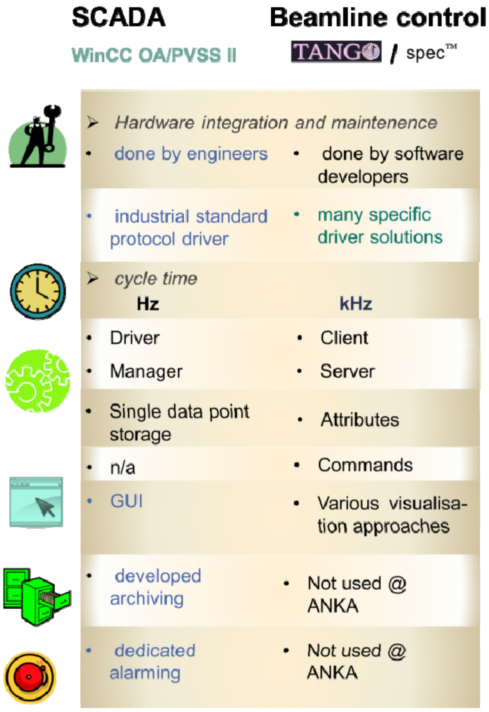
The beamline control at ANKA was originally fully based on SPEC and is currently moved to TANGO [2] and SPEC [3] which has been fully integrated by expanding by TANGO server capabilities. This approach implies the essential need to provide a stable and fast link, which does not increase the dead time of a measurement to the slower WinCC OA SCADA system. The open architecture of WinCC OA offers a smooth integration in both directions and therefore gives options to combine potential advantages, e.g. native hardware drivers or convenient graphical skills.

The implemented solution will be presented and discussed at selected examples.

CHALLENGES

An important factor for the scientific excellence of a synchrotron is based on its beamlines and their rapid and precise measurement instrumentation and data collection. Obviously a huge number of specialised devices is to be managed. TANGO turned out to be an appropriate tool regarding the speed and the required flexibility for the experimental instrumentation. At the same time and time scale essential environmental information needs to be provided by the SCADA system.

The initially much slower SCADA provides at smallest engineering cost level the required long term stability and scalability and guarantees the robust control of PLCs and any house keeping equipment. But the different event execution times of WinCC OA in respect to TANGO requires an adoption of the data processing.



SCADA	Beamline control
WinCC OA/PVSS II	TANGO / spec™
➢ Hardware integration and maintenence	
• done by engineers	• done by software developers
• industrial standard protocol driver	• many specific driver solutions
➢ cycle time	
Hz	kHz
• Driver	• Client
• Manager	• Server
• Single data point storage	• Attributes
• n/a	• Commands
• GUI	• Various visualisa-tion approaches
• developed archiving	• Not used @ ANKA
• dedicated alarming	• Not used @ ANKA

Figure 1: Comparison chart of basic features of WinCC OA and Tango based beamline control. Key benefits of the SCADA and TANGO used at ANKA are marked blue.

Both control systems show a certain analogy but at the same time different fundamental approaches. WinCC OA is set up on central event manager(s) which guarantee the consistency of the data, generate events, derived process values, and a bunch of other management functions. In contrast TANGO offers a clear client-server-concept. It results into a flexible and fast control structure which doesn't stand out ab inito as a clear interaction scheme or presentation concept.

Due to the fact that none of both systems may substitute the advantages of the other one a seamless cooperation needs to be sought.

SOLUTION

Both systems set up on TCP/IP communication and offer a complete API for Windows and Linux. The manager/driver concept of WinCC OA is comparable to TANGO server/client pair. Hereon set up the approach to combine manager – server and driver – client pairs.

At ANKA Linux was chosen as the main platform and all mangers have been designed to this environment. But there are no principle limitations to that operating system.

*thomas.spangenberg@kit.edu

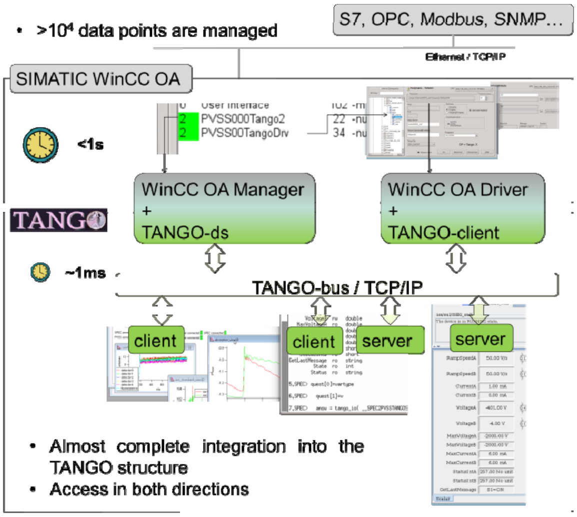
The WinCC OA driver is in relation to TANGO a client which connects by multiple standard TANGO proxies to the servers. Polling and event based data point connections may be realised by the driver. The productive version at ANKA is based on polling. Some minor data type adaptions are done on driver level.

The TANGO server implementation for WinCC OA permits a straight forward read and write access to the WinCC OA data set. Supported data types of the current version are, integer, boolean, double, and string.

A WinCC OA data point and its addressed element are well described by its name which is a unique string and it is used as an index for the cache. Type and name are used to connect to the WinCC OA event manager and the requested value will be cached. Any change of it will update the cache.

Any update of a value is done in the comparable slow WinCC OA time scale. But a request to a successfully connected data point is handled by the TANGO server interface and is defined by the speed of C++ map. The response time of these maps is considerable shorter than 1ms and scales logarithmically to the size of these objects. In case of a higher workload the application is free scalable via the number of the server applications. Additionally the load to the WinCC OA event manager by multiple access of the same value from different clients is reduced by a cache mechanism, which effectively prevents the single threaded WinCC OA event manager to be blocked by an excessive number of client requests.

The write access to WinCC OA is indirectly feeding an input queue to the event manager. Again, timing issues needs to be considered. Requests from TANGO clients are queued and handled sequential with WinCC OA manager speed.



• >10⁴ data points are managed
S7, OPC, Modbus, SNMP…
Ethernet / TCP/IP
SIMATIC WinCC OA
<1s
User interface
2 PVSS000Tango2
2 PVSS00TangoDrv
TANGO
WinCC OA Manager + TANGO-ds
WinCC OA Driver + TANGO-client
~1ms
TANGO-bus / TCP/IP
client
client
server
server
• Almost complete integration into the TANGO structure
• Access in both directions

Figure 2: Schematic overview of the WinCC OA / TANGO interaction for beamline operation. The intrinsic time scale difference is managed by the developed WinCC OA manager and driver.

TECHNICAL CHALLENGES

Both systems are offering a complete C++ API which compiles each separately with gcc 4.x compilers. But the standard makefiles claim different compiler options and libraries. Additionally the WinCC OA is limited to 32bit.

Due to incompatible header requirements a direct exchange of data types or classes doesn't work. Dedicated data exchange classes were developed which are connecting the WinCC OA and TANGO part by a bidirectional data queue and granting the thread safe access to the exchanged data. There are different approaches for the TANGO server and for the TANGO client.

A major problem of combining a TANGO server and a WinCC OA manager is that both are claiming to define the int main() originally. Even if the source is available and the name conflict might be resolved it turned out that at first the TANGO server needs to start up and the WinCC OA manager template part is possible to run in a separate thread with a renamed and modified main() function. As a third argument the address of the data exchange is passed for a separately compiled library which contains the complete WinCC OA manager.

For compatibility reasons of the by POGO generated TANGO server the WinCC OA part parameters had to be defined as TANGO server property and are forwarded to the WinCC OA thread. To handle the TANGO server as a 'normal' WinCC OA manager, it was wrapped by a script which overcomes the difficulty of setting natively unknown command line TANGO parameters to a WinCC OA application by its management tools.

The combination of TANGO clients and WinCC OA drivers shows smaller difficulties. The program was developed as a standard WinCC OA driver project. The TANGO client properties are provided by the standard TANGO class and operated in an own thread. The same queue classes of the manager are used to create a thread safe data flow. Due to its original WinCC-OA nature the driver is also managed by the WinCC OA management.

ACTUAL STATE

The server (manager) as well as the client (driver) are in production state and hosted on a VMware ESX server as a small 32bit SuSE Linux 1GB RAM virtual machine.

The server gives a response of less than 1ms request to a request of a previously 'connected' value. Scalars of the type bool, int, double, and string are implemented in the current version. Other types will be implemented on request.

The client operates as a native WinCC OA TANGO driver and permits a standard WinCC OA data integration with the only constraint that it is currently not possible to browse the existing tango attributes of a device server.

CONCLUSION

The combination of the SCADA system WinCC OA and TANGO is fruitful for both sides. The TANGO world gains a well-engineered alarming and logging and a bridge to industrial protocols. Many industrial devices are now available to TANGO without any extra driver development. Due to caching any WinCC OA data can be accessed without any time loss in the timescale of the typical TANGO measurement processes. On the other side WinCC OA application range with its easy and straight forward to develop GUI is significantly extended for scientific applications by the huge number of already developed TANGO device servers and it's attributes and simple commands. And last but not least the WinCC OA TANGO driver allows automation engineers a seamless access to TANGO servers without special TANGO knowledge and general software development skills. The load of GUI development for standard automation tasks is shifted from software developers to engineers and therefore reduces the overall engineering costs and frees resources for more complex software tasks.

The solution is in productive state at ANKA.

REFERENCES

[1] http://www.etm.at
[2] http://www.tango-controls.org
[3] http://www.certif.com