

What to Do About Incomplete Requirements?

HOW TO AVOID THE STALL OF A PROJECT BEFORE IT HAS EVEN STARTED

BY: MARK PLESKO (COSYLAB)



It has happened finally – your new large accelerator project has been approved. After years of waiting, with the initial design papers long obsolete, the government has secured the funds. Now is the time to party! Actually, it's the only good time to party, because it will never be as good as it is now. Once the project has started in earnest, the problems will be arriving too. And since the end of the project is never as clear, if there is an end at all, there will never be a more clearly defined moment in the lifetime of the project. Apart, maybe, from the day the minister visits the facility and presses the red button.

We, the control people, are usually still quite relaxed in the beginning phase of the project. The control system is never on the critical path, so there is no need to start early and, furthermore, we don't have requirements yet, have we? Ah, requirements, the magic word that makes the blood of some people boil and other freeze to death. The word of both the brave and bold, who use it to demand ("give me your requirements"), and the shy and humble who use it for excuses ("we haven't the requirements yet").

So, what should we as controls people do? Should we rejoice with the others and start working on prototypes, technology evaluations, etc., or should we return to our dull daily business and wait for requirements to slowly trickle in? Well, it depends solely on which of the above two options we really want to happen. Requirements can be essentially a good excuse to do any of them. Unfortunately, they can be a good excuse also for others.

It is a fact that the control system group rarely gets proper and clear requirements. And when the other groups finally come knocking on our door, they behave as customers - they want already the final product, as if they were shopping products and not buying development. Knowing that this is a fact of life, what can the control group do to preempt such embarrassing situations?

The obvious solution is to just keep asking for requirements from early on. However, as it hap-

pens only too often, the obvious solution is not the best. Even in the Middle Ages, when they could still employ torture, it usually got them the wrong answers. With that option gone, there is little leverage that we have over other groups. And if we keep insisting and threatening, we just make everybody hate us, with no other tangible success.

Based on my experience, it is my opinion, although I am aware that many people may not share this opinion, to start developing the new control system even with a limited or nonexistent set of requirements. There are several good reasons:

1. No complex development ever follows the simple "waterfall" scenario with the phases like requirements, design and implementation following linearly one after the other only once the previous is finished. In reality, some requirements are dropped during the way, new come in, and what's more, development cycles often follow a spiral and pass several times through all the phases. So, why not start the development with what we have at the moment? Maybe make a "straw man" design [1] or reference implementation and use the newly gained experience in the next round of the spiral?

2. Developing a vertical prototype early on in the project creates immense motivational force to the control team. Technologically, it creates a sandbox to develop and test new concepts and technologies and psychologically, it creates awareness in the team and gives the junior members of the team a chance to profile themselves.

3. If the list of requirements is long, people often forget that requirements have different weight, different importance, and that the importance often depends not on technical considerations but on political power of the person that has provided them. Most of the politically sponsored requirements

don't have a "price tag" attached to them – they're just somebody's fancy. By having done a prototype, even against different requirements, we're in a much better position to estimate the cost of implementing incoming requirements and to argue against exaggerations; or ask for a larger budget and more people.

4. Seeing a physical thing makes people think much more: a customer of ours just recently gave a new board that we have developed to a group in his project that he thought they will use. They borrowed the board for one day and only after careful inspection found out that they want a different connector. Surely they could have specified this earlier? Fortunately, this was a prototype and it is no big deal to add another connector.

5. The other use of a vertical prototype is that people can actually use it already for their work, e.g. a power supply control prototype can be used in magnetic measurements, beam instrumentation people can use the controls framework to develop their own software, and so on. Thus we make sure that people get used to the new control system early on, even though it doesn't exactly match all future requirements.

6. And last but not least, there is the fun factor: wouldn't you rather play with (call it investigate) new technologies and develop prototypes? Just don't tell it to your boss: giving him all the other previous reason should be convincing enough.

But then you may prefer the safety of your daily chores. In that case, simply reply: "I'd like to start the new development, but we just don't have the requirements yet".

[1] Straw man, http://en.wikipedia.org/wiki/Straw_man_proposal

Control Systems For New Large Experiments

BY: ROK SABJAN (COBIK*), MIHA RESCIC (COSYLAB)



INTRODUCTION

During the years control systems have evolved, together with information technology, computer science and electrical engineering. In the starting days, little equipment was available off-the-shelf. A handful of physics labs with difficult requirements were just not commercially interesting. At that time engineers were also scientists. Today, big number of experimental projects and the advance of computing allowed widespread standardization of components. Nevertheless, new experiments are still hiding more technical challenges and questions.

STANDARDIZATION IN LIGHT SOURCES

Plenty of light sources were built in the last decades and they have a lot in common with respect to the control system. Control system packages have matured through collaboration and the increasing market has attracted industry. This has reduced control system budget from 10% to 5% of the machine's budget. The challenge today is to implement a control system with state-of-the-art technology on a smaller budget, shorter time-frame without sacrificing quality.

PUSHING THE LIMITS OF CONTROL SYSTEM COMPONENTS

Machine Protection System

Reference implementations of a flexible machine protection system (MPS) do not exist yet. Based on collected requirements from several projects new features of MPS are required. One such is the reconfiguration of the MPS depending on the working mode of the machine. Integration with the timing and control system is highly desired, allowing quick reconfigurations and post-mortem analysis. The new design allows responses in the range of microseconds making the speed of light the biggest constraint.

Hard Real-Time Feedback System

Implementations of a hard real-time feedback system already exist but are largely based on proprietary technologies and specialized solutions. We measured the deterministic performance of Gigabit Ethernet as a task for ITER. Our results showed that we can already achieve a very good latency of 0.5ms for data rates that are typical for accelerators. With 10 Gigabit or even faster Ethernet, we are confident that Ethernet is a very good choice for development efforts.

Timing system

New complex machines require new complex timing systems. New features like virtual accelerators, timing super-cycles and event acknowledgements are introduced. The existing (off-the-shelf) timing solution cannot provide all the needed functionality, but they can be used as the basic component, the transmission layer. Still, a lot of customization work is necessary. You can purchase the transport layer, whereas the application layer is machine specific and needs to be implemented for every project individually.

COMPLEX COMPONENTS AND INTEGRATION

We have established that there are definitely trends towards standardization of control system components. But, components are getting more complex and they require more time and effort for integration. Choices need to be made early in the project which is risky if not all aspects are considered.

Basic control system package

Traditionally, the first choice is about the control system package itself. In fact, we believe that people decide for a control system package in a similar way as when they are buying a car: we decide based on emotions and later we rationalize this discussion with architecture description and features. Luckily, most of control system packages are mature and modern technology will enable you to finish your project whatever your choice might be.

Integrating other packages

Many facilities use more than just one control system. Another set of examples is the machine physics world. Usage of many different packages is to be avoided, if possible. For the remaining case, we believe that responsibilities and requirements for the interfaces must be well defined. Documentation and maintenance of the systems must be considered. Technical problems come second to interpersonal relationships. It is usually best to adopt best practices developed and lessons learned by a previous similar experiment.

Distributed development and 'in-kind' projects

New large experiments, such as ITER, FAIR or ESS are often started as international projects with many in-kind contributions. This requires very rigid standardization. Every year, the ITER controls group publishes the Plant Control Design Handbook (PCDH), which describes all the standards, and releases the Core System

software, the set of standard, ITER approved community tools and software drivers. The standardization continues with the main architecture, hardware platform and IO boards, operating system and software packages. Project life-cycle, naming convention and test plans are also specified.

FOCUS ON DEVELOPMENT PROCESS

Building a complex system from many components is an engineering task but control system development has an even more complicated cycle: write specifications, architecture, design, prototyping, define test procedures, implementation, write documentation, testing, debugging and acceptance at customer.

Projects are increasingly aware of the development processes so more things require focus. The signal list is a golden list that represents the contract between different subsystems and developers. It heavily relies on a good naming convention. Logistics of installation and error handling should be considered, time needs to be planned for testing and debugging. Overall system responsibility should be kept in-house when control system parts are outsourced.

CONCLUSIONS

Standardization is the key trend emerging with new and complex projects. Today, integration replaces development as the biggest aspect of a controls project. How will all the components fall into the main architecture, what will be the interfaces and how the requirements will be addressed, are the main questions. Because of increasing organizational risks the focus needs to be shifted to more strict definition and implementation of development processes and rigorous standardization.

In short, control system development is becoming more and more an engineering discipline and less like a science.

* The Centre of Excellence for Biosensors, Instrumentation and Process Control - COBIK, Velika pot 22, SI-5250 Solkan, Slovenia COBIK is an operation financed by the European Union, European Regional Development Fund and Republic of Slovenia, Ministry of Higher Education, Science and Technology."

Efficient Access to Timing System's Time in Linux User Mode

BY: KLEMEN ZAGAR, JANEZ GOLOB (COSYLAB), ANZE ZAGAR (COBIK*)

Dedicated timing systems are designed to provide an accurate notion of current time to distributed nodes, and there are several ways in which a node can make use of this accurate time. Ideally, the time is handled directly by the hardware – e.g., timing receiver card triggers data acquisition via a digital signal. This ensures very accurate synchronization of the data acquisition with accuracy in the order of one period of the timing card's clock (for a timing system with a base frequency of 125MHz [1] this would mean 8ns).

Many times, however, integration of timing and other activities at a node is infeasible to do directly in hardware, and needs to be done with the help of the CPU. If the CPU is running the Linux operating system, the only place to access timing card is in the kernel. If developing (and maintaining!) a kernel module is feasible, that's how it should be done. But if one needs to remain in user mode, the overhead associated with user-kernel mode switch must be taken into account.

Whereas the kernel-user mode switch is not expensive *per-se* (this is essentially triggering of a software interrupt via the *syscall* mechanism [2]), the necessary logistics surrounding it can add up. For example, one might leverage the *ioctl* or *read* system calls to obtain the current time from the timing card, but these require operations with a character device driver, and has a cost in the order of several microseconds. In addition, retrieval of the time from the timing card to

the CPU can be costly, too, as it needs a transaction over the bus.

Can acquisition of current time in user space be done with less overhead?

The idea is to use the CPU's clock, as given by the x86 RTDSC (»Read Time-Stamp Counter«) instruction, as follows: let's assume that at some time t_0 , both the value of the RTDSC counter and timing card's time were known. Let's denote these counters as $C(t_0)$ and $T(t_0)$ respectively. Let's further label the frequency of the RTDSC counter with ν_c , and frequency of the timing card with ν_T . Then, at some later time t , the values of the counters are:

$$C(t) = C(t_0) + (t - t_0) \cdot \nu_c$$

and

$$T(t) = T(t_0) + (t - t_0) \cdot \nu_T$$

Given $C(t)$ (i.e., the value given by RTDSC), we can infer $T(t)$, which is the timing card's time. Considering that the above is a linear system of two equations and two unknowns, after eliminating $t - t_0$, the value of $T(t)$ can be expressed as:

$$T(t) \approx T(t_0) + \frac{\nu_T}{\nu_c} \cdot [C(t) - C(t_0)]$$

We have deliberately put the »approximately equal« sign above, because of the following reasons:

- Due to temperature instability and other external factors, neither ν_c nor ν_T are

constant, and can drift even a few microseconds per second.

Additionally, synchronization algorithms that make sure that timing card's clock follows the reference might prefer to adjust the frequency ν_T .

We have implemented a user-mode function that determines the approximate time using the equation above. On our particular platform (2,66GHz CPU) it took about 50ns for this function to complete. Adding 50ns of timing card's inaccuracy [3], the accuracy of the time obtained in such manner is in the order of 0,1ms.

[1] Micro-Research Finland: Timing System, <http://www.mrf.fi>

[2] man page for *syscalls*

[3] National Instruments: GPS, IEEE 1588, and IRIG-B Timing and Synchronization Modules for PXI and PXI Express: http://www.ni.com/pdf/products/us/cat_pxi_6682.pdf

* The Centre of Excellence for Biosensors, Instrumentation and Process Control - COBIK, Velika pot 22, SI-5250 Solkan, Slovenia COBIK is an operation financed by the European Union, European Regional Development Fund and Republic of Slovenia, Ministry of Higher Education, Science and Technology."



Cosy-shirts over the globe - photo contests!

Hi. Here we are again with a new picture of a lucky Cosy T-Shirt owner. Mark Wagner from Gamma Vaccum is one of the readers who sent us a picture showing how proud he is to wear it. You will notice by the expression on his face that he was very happy to do so. We think one of the reasons he is so happy is that he knows he will have another Cosylab T-Shirt for his collection.

If you would like to get a new Cosylab T-Shirt then you also can

too. Send us [1] a picture of someone wearing a Cosy T-shirt and you can also get your own. We will reward the best and funniest photos by publishing them in the newsletter and by sending you another Cosylab T-shirt of your choice.

[1] mail to: controlsheet@cosylab.com



Mark Wagner a proud owner of new Cosylab T-Shirt

Power Converter Controller for MedAustron

HIGH PRECISION STEERING FOR CANCER KILLING ION BEAMS

BY: ROK TAVCAR, LUKA SEPETAVC, (COSYLAB), JOHANNES GUTLEBER (CERN),
ROK STEFANIC (COBIK*)

A new synchrotron [1] is being built in Wiener Neustadt, Austria by EBG MedAustron GmbH. It will use protons and carbon ions for cancer treatment and for medical and non-medical research. Trial operation is estimated to begin in 2013.

Cosylab has been awarded a framework contract to provide development services and skilled manpower to the MedAustron team for control system design and development. This development project is carried out at CERN (Geneva, Switzerland). Currently, our joint effort is focused on the accelerator main timing system, power converter controllers, front-end embedded software and integration of numerous industry-supplied subsystems with the supervisory control system.

The particle accelerator comprises approximately 300 magnets used for focusing and steering particle beams around the ring. Beams can be generated from various ion-sources and are accelerated to injection level via a pulsed LINAC. Acceleration to energy levels needed for irradiation is performed in a synchrotron. When a beam has reached the required energy level, it is extracted from the synchrotron into the high-energy beam transfer line from where it is delivered to individual clinical irradiation or experimental rooms. Accelerator synoptics are shown in Figure 1.

Magnets in the synchrotron ring are powered by purpose-built, voltage-source type power converters. They have to deliver precisely configured amounts of power at defined moments in time and must be operated synchro-

nously within a time frame of $1\mu\text{s}$. The distributed system in charge of operating all magnets highly synchronously is the Power Converter Controller!

Each individual Power Converter Controller (PCC) connects to about 30 power converters and tells them how much power to output. A supervisory control system configures the PCCs to "play" pre-defined waveforms to power converters. All PCCs are closely coupled with the accelerator main timing system which precisely sets the pace of PCC operation and assures system-wide synchronicity.

GEEK FEED – PLATFORM AND TECHNICAL DETAILS

Each PCC consists of a National Instruments PXI crate housing a PXI controller, a Micro Research Finland's (MRF) timing receiver card and several programmable logic cards - FlexRIO FPGA Modules. Each of the latter has a FlexRIO Adapter Module (Figure 2) with six fiber optic links for connecting to Front End Devices (FEDs). This device has been designed and built by Cosylab for MedAustron. The FlexRIO cards take care of real-time tasks and deterministically transmit sequences of power converter setpoints via real-time fiber links to FEDs, which talk directly to power converter regulation electronics.

MedAustron has chosen to design in-house a generic DSP-based digital/analogue regulation board that can be connected to many voltage sources on the market, thus easing the pro-

curement process, clearly separating responsibilities and risks and allowing for precise performance comparison of the offered power converters.

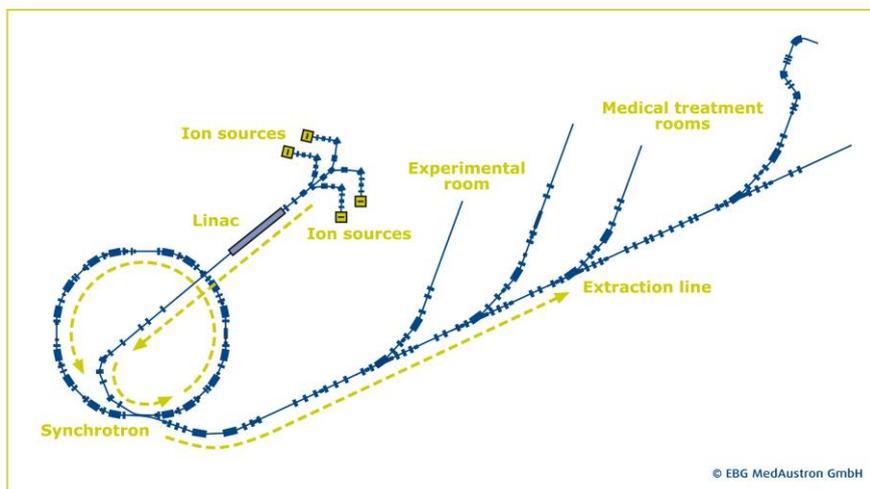
The PXI controller runs the LabVIEW real-time operating system with software support, commanded by the supervisory control system, which also delivers setpoint sequences to be played to FEDs. The link between FlexRIO cards and FEDs is based on a custom real-time protocol which



Figure 2: Custom developed FlexRIO Adapter Module with 6 fiber connectors

guarantees deterministic delays for high priority traffic, while allowing transport of lower priority data. The protocol is fully customizable, providing great flexibility.

The FED is a generic FPGA board (Figure 3) developed by Cosylab for MedAustron. It forwards the data received over the real-time fiber link to a power converter and concurrently acquires measurement data from the power converter. The board's IO capabilities are highly configurable, thanks to a modular, pluggable design. Examples of interfaces that are currently used are RS232/422, a SCSI-5 parallel bus interface and coaxial trigger outputs. FEDs act only as interface adapters and do not interfere with the traffic that >>



© EBG MedAustron GmbH

>> passes through them. Several different types of power converters will be used at MedAustron. FED's configurability allows integrating power converters with different electrical (RS-232, 422...) and logical interfaces (different command sets), all controlled by a single system. If you need to add a new type of power converter with an exotic interface, you just have to create a new IO port adapter extension that is plugged into the FED - the PCC remains virtually untouched!

AWESOME! IS IT ALREADY FINISHED?

As of December 2010, a first series has been produced to evaluate its robustness in CERN's existing accelerator test-beds. Basic software support to carry out functional tests and to evaluate the boards' maturity has been developed. Together, they demonstrate the entire vertical col-

umn of the system – support software, FlexRIO cards with adapter modules and the fiber link to FEDs. The system is able to successfully demonstrate deterministic transmission of high and low priority data over the fiber link. If you disguise yourself as a power converter and connect to the FED via a terminal emulator, you see power setpoints playing in tune with the timing system.

We probably don't have to mention that this year's Christmas lights at Cosylab are blinking in perfect synchronicity. :)

[1]: MedAustron web site: <http://www.medaustron.at/index.php?lang=english>

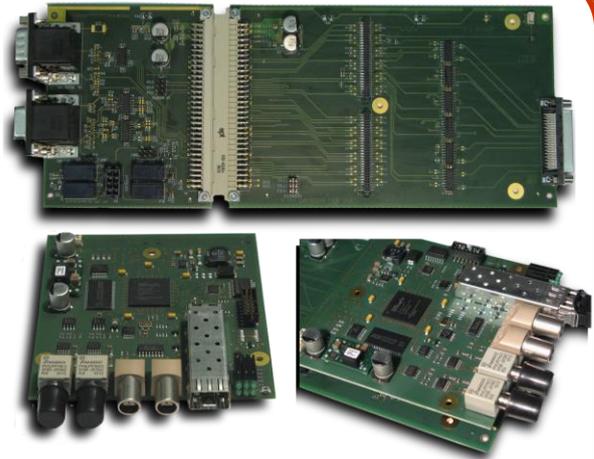


Figure 3: Top: base board, bottom: FED & baseboard

* The Centre of Excellence for Biosensors, Instrumentation and Process Control - COBIK, Velika pot 22, SI-5250 Solkan, Slovenia COBIK is an operation financed by the European Union, European Regional Development Fund and Republic of Slovenia, Ministry of Higher Education, Science and Technology."

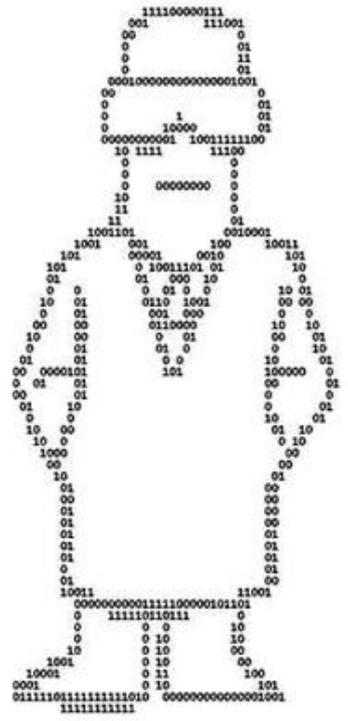
Switch on 2011— MERRY ZEROES AND ALL HAPPY ONES IN 2011

Hello everyone, on such special days coming up, Cosylab would like to wish everyone very happy holidays and a great new year 2011. To do this, we would like to send you this very special animated New Year card, much in our Cosylab particular style. Please click on the picture of the funny looking scientist to the right of this text to view it.

And since this is a time for giving, we would like to give a special gift to the first 10 lucky readers who see our New Year card. But of course, this gift will not come without any effort on your part. In order to be eligible for this gift you must first solve our secret puzzle! On the picture on the right you can see a secret binary code in red letters forming 2011 underlined. If you can retrieve the code reading the lines horizontally, break it and find the secret message behind it, send it to our email controlsheet@cosylab.com. We will randomly select 10 correct answers and the lucky ones will be rewarded with a special Cosylab gift. (Hint: the secret code might be somewhat inspired by ASCII.)



011101000 111001 #1101 00101
 381 ###1 1100 130 #1011010 10101000
 01 000 #00 11 00 00
 01 001 10 00 01
 011 01 18 00 11
 110 11 01 11 00
 110 110 111 10 10
 000 110 010 08 00
 0110 011 #11 18 11
 1100100### 111001001 381010110111 0 #110111 #1101
 1110100000 0110010 #11101010110 1111011000##11
 1000## 1000001100001001##00110110101100##1 00100###1001001



COSYLAB d.d.
 Teslova ulica 30
 SI-1000 Ljubljana
 Slovenia

Phone: +38614776676
 Fax: +38614776610
 E-mail: controlsheet@cosylab.com
 Web: www.cosylab.com

visit us on http://cosylab.com/solutions/particle_accelerators/control_sheet/