From WP 7 spec:
Objectives: Both the global control and monitoring of the whole facility and the low-level control and monitoring of the thousands of distributed antenna elements within the individual arrays place large demands on the support systems and software.

I don't think the demands are all that overwhelming from the point of view of control software or the amount of computing required. Assume say 100 numbers per element, 10'000 elements, accessed/updated once per second. This implies sustained data rates of a few magabytes per second. This is only a trickle compared to the anticipated data flow of the actual measurement data. A single central workstation easily handles the control data if the data can be fed to it efficiently.

This work package includes an evaluation of existing concepts and strategies for managing real-time, geographically widespread systems [...].

At this time, I see little value in that kind of effort, apart from satisfying a considerable academic curiosity. This is mostly because, on the down-to-earth level, there exists an overarching practical boundary condition on the C&M system.

- The 3D C&M software ultimately needs to be maintained and operated by EISCAT, even in the case that much of the 3D hardware would be maintained by some commercial consortium. No such consortium will ever, I assume, be running our experiments for us.
- The C&M software is crucial for the usability, hence the ultimate success, of the radar system. Therefore,
- It is necessary that EISCAT has good in-house understanding and complete in-house control of the C&M software, so that the software can be adapted quickly, cheaply, and without fuss, to whatever will be required for the hardware on one side, and for fulfilling the needs of the user community on the other side. I do not want to start filling web forms of some commercial service contract to get some less-than-perfectly specified user wish granted before noon tomorrow. I freely admit that our response has, in reality, not been on a 24 hour basis. But it took less than two weeks by two of our staff to design and implement from scratch into EROS the capability to handle an extra receiver at ESR, requiring extensive expansion of the internal EROS variable set, but saving an experiment campaign. Significantly, not much patching has been required afterwards in that part of EROS. Thus,
- I think that all essential parts of the C&M software must not only be designed but actually implemented by EISCAT staff. Even if a multisite radar C&M system would be available somewhere—which I somewhat doubt—I would still be hesitant about using it.

EISCAT has not much staff, and has even less available for software development. This suggests to reuse what software we have, especially, the EROS radar operating system. For more than 25 years now, EROS has shown to be a reasonably efficient and flexible means to operate our ever-developing radar hardware, both in maintenance and during experiments. I will argue in this note that EROS does not need drastic changes to be able to handle the 3D sites on the same footing as it handles the other parts of the EISCAT empire.

One of the few explicit aims in EROS development has been to provide a uniform interface for the radar hardware across all EISCAT sites. The purpose has been that from a user point of view, all our radars look and feel as much the same as possible, and feel part of a single system. This should be the case also in the future, only, to reign-in on the increasing hardware complexity, more so. An increased number of antenna elements and the extra degrees of freedoms involving multiple beams notwithstanding, the 3D system is still "just" a multi-static radar: transmitters, antennas, receivers, characteristic data flows. The EROS system has from its beginning been tailored to handle such systems. I do not anticipate much qualitatively new required for the new C&M system.

I do not view adding the 3D sites to the EROS drastically different from incorporating the P site (ESR plasma line receiver), the D site (space debris receiver) or the H site (heating system). Especially, I expect that the same proven software development strategy can be followed for the 3D sites as has been followed for the existing sites. The bottom-up strategy involves four steps.

1. Engineering requirements alone decide what hardware there will be. The hardware magically comes about, with some computer interface.
2. Some preliminary commands and scripts are developed by the engineering teams.
3. An EROS command-line interface is built based on those by-then relatively well-tested engineering commands.
4. Possibly some higher-level EROS commands are constructed as a convenience for experiment programming.

Some iteration of steps 2-4 may take place, for it is of some significance to have a good, natural set of EROS commands. Just as there are good and bad GUIs, there are good and bad command-line interfaces.

Much of the interesting hardware nowadays will be computer controllable by default, and I count on some kind of engineering computer interface to be available for the 3D hardware. The interface may perhaps be provided by the project engineering teams for their development needs, or by a commercial equipment vendors, but is anyway provided outside the present work package. Then from the EROS point of view, and hence from the point of view of WP7 at this stage, it does not matter much what the actual control parameters are, or how numerous they are.

However, it has been a simplification in the existing EISCAT system that much of the hardware has been rather efficiently accessible via a UNIX command line interface. It would be inconvenient and possibly disastrous if this could not be realized for the 3D also. Underlying the command line interface there must be a C-programming interface. This is a basic requirement, and pretty much, our only requirement: building an EROS-style C&M system requires that the hardware be fully and efficiently accessible with C from a UNIX machine. What I especially do not want is to find some commercial graphical programming system under Windows standing in the way of EROS access to the 3D hardware.

The number or nature of the hardware control parameters does not matter much from the EROS point of view, but what does matter is on what time scale, with which timing accuracy, and how often the hardware must be accessed. However, I do not anticipate really new needs on the 3D system: already, EROS handles two distinct classes of needs, characterized by the required timing accuracy of the controlled events: the microsecond time scale and the one-second time scale. Essentially, EROS itself handles only the one-second subsystems and tasks. For the fast subsystems, EROS only arranges for triggers to be generated; the subsystems are then started and operated autonomously via the radar's high resolution and high accuracy time-and-frequency subsystem. This arrangement allows such a big simplification in the RT-needs of the C&M system that we should try to stick to it. If this can be done, I do not see an obliging need for a "real" real-time system at the 3D sites any more than at the existing sites.

The twofold separation of control tasks has made it possible to build EROS as a lightweight collection of loosely coupled processes, interacting mostly via simple text messages. This has simplified the initial programming but especially the later developments, as it obviously is safer and easier to modify a loosely coupled system than a tightly coupled system. One aspect of the lightweightness is that EROS is not deeply entangled with the UNIX host operating system. EROS is just another ordinary user-level program. If it crashes, it just needs to be restarted and nothing much happens in the hardware, though the experiment will need to be restarted to get the EROS internal state up-to-date. EROS has been able to take this care-free approach, because it has been possible to trust that there are enough hardware precautions, interlocks, so that a software glitch cannot do serious damage to the hardware. As the 3D hardware is being planned for unattended, remotely monitored operation, I assume that this continues to be the case also in the future.

The tcl/tk language system underlying EROS has been originally designed for this kind of bottom-up, incremental development of device control, and is quite helpful in that task. The open-source tcl/tk language is now more than twenty years old, well established and widely used, and still in active development (current version is 8.4.14) . The language has a java interface in addition to the C-interface; its GUI facilities are used also for the GUIs in other languages such as python; and the system is supported in all major platforms. The tcl/tk programming language is as solid and future-proof foundation for the C&M system as one can hope for.

There has been several significant improvements to the tcl/tk language since the present version of EROS, EROS4, was initially implemented about ten years ago, such as namespaces, threads, and native support of non-string data types. Even at the existing EISCAT sites, EROS would benefit of these new features. Moreover, EROS should be strengthened internally in several ways to make it more robust and even more easily modifiable. Still, these are only technical issues, not such issues of principle that could make or break the gradual incorporation of the developing 3D system under EROS.

But are we lacking something more fundamental in the present EROS, something else than configuration parameters for new sites and extra commands for the new hardware, before we can handle the 3D system?

I. We will need a more universal and comprehensive remote access to the radars from "anywhere", both to the 3D sites and the existing sites. This appears not too difficult to provide, and will be a feature of the next major version of EROS anyway. It should be remembered that for the 3D system, we will need to provide remote access also for the "engineering" data, which in the present system are visible only locally at the sites, and often only via a GUI.

II. EISCAT radars are flexible, programmable systems, due to a large part to the input of the EISCAT community. It is difficult to anticipate the future requirements of the community. We therefore intend to make the EROS system more open, more programmable in the future. For special needs users, including us, must be able to develop fully-powered EROS extensions—packages they are called in tcl/tk—that can access the internal routines and state variables and the multiple processes of EROS, so as to allow programming experiments that are more interactive and more data-driven than the present experiments.

III. It is a design requirement to be able to change the system data flows based on special conditions. This clearly is related to (II). It would be nice if some general framework for this could be found, but I'm skeptical. The EROS kernel probably will provide some new scheduling primitives in addition to the present start and stop experiment. We might want to interrupt a running experiment to launch a new one, and when done, return back to the previous experiment. But even this seemingly small addition has problems if pursued too far. How deep branching-away from the initial experiment should be allowed? Should we have an experiment stack? When should the stack be flushed? By what authority? We need to be realistic about what can be provided in the general case, and what is perhaps better handled via special cases only.

IV. We will need ways to make sense of the largish amount of system data related to the antenna arrays. Some C-level programming will probably be needed to handle the C&M data efficiently, and some display graphics will undoubtedly be needed. At some stage, some kind of monitoring GUI may become an attractive option. I myself tend to avoid GUIs until they are *really* unavoidable, for GUIs often discourage programming. When you have a GUI, you tend to do your repetitive work interactively over and over gain, when you really should implement a new command via a short script. On the other hand, graphical display of large data-sets is a difficult subject, and shopping around at the world's large-scale data-processing facilities for ideas, and maybe even for packages, could turn out to be useful.

V. We need to be more careful in the future than we have been so far in ensuring that the relevant radar state information will be included into the saved data streams.

As perhaps is evident from the above list, I am not capable of finding any strong reason why we could not continue to use the EROS system also for the expanded EISCAT system. We will soon start introducing the 3G test array as a new site to EROS. My intention is that most if not all of the wish list items (I-V) will be addressed at that time already (this spring), at least to some degree.

What about the relation to other 3D work packages, how should WP7 interface to them? My feeling at this time is that the bottom-up software development procedure is robust enough so that for us "anything goes", as long as *some* UNIX C-interface to the hardware (and DSP) will be ultimately be available. What that interface more precisely is we do not need to know anytime soon. Once the C&M infrastructure is in place, just adding commands is easy. Within the above boundary condition, the other packages should feel quite free to define their C&M interfaces in the way they find convenient; if the developers can access their developing systems, so will the future EROS also.

In summary, I can endorse the part of the management summary of the 1st 3D annual report on WP7 which says that It is *expected that the EISCAT_3D control and monitoring systems will grow out from, and largely replicate, the EROS system currently in use at EISCAT.* Perhaps I'm viewing the 3D more as an integral addition to an (updated) EROS than a separate replica, but this is mostly semantics. The important point is that it seems possible to move from here to there rather smoothly, gradually, iteratively, and without too large an overall effort. In fact, much of the work will benefit the existing EISCAT sites also, and should probably happen irrespective of whether the full 3D system will someday be built or not.

30-Jan-2007

Jussi Markkanen
jussi.markkanen@sgo.fi
EISCAT Sodankylä site