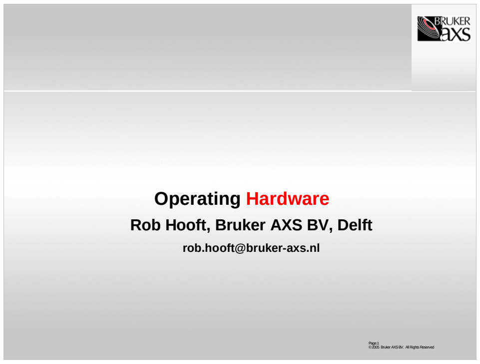
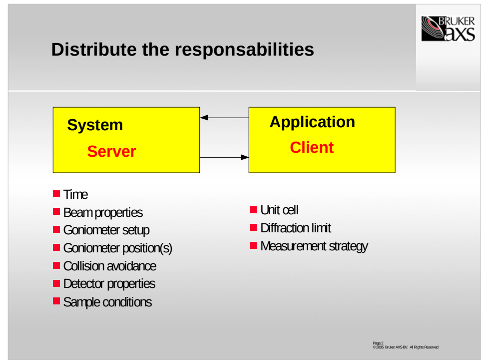
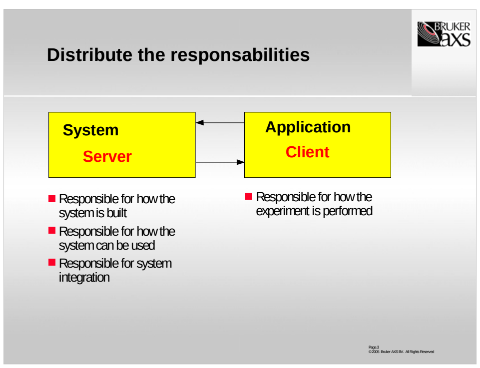
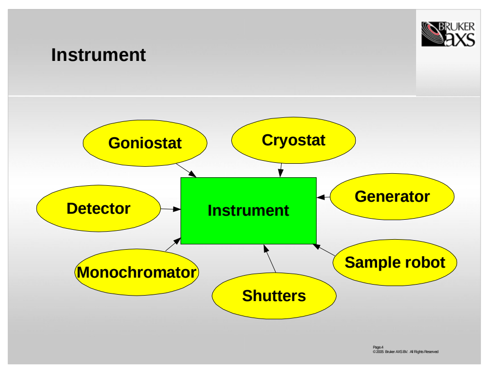




We have seen this image earlier, there with the intention of making clear that the application should not deal with instrument parameters, but be able to deal with any set of relevant instrument parameters. Now we look at it from the other side: the server system will need to take care of all of the hardware-dependent issues of a crystallographic experiment.





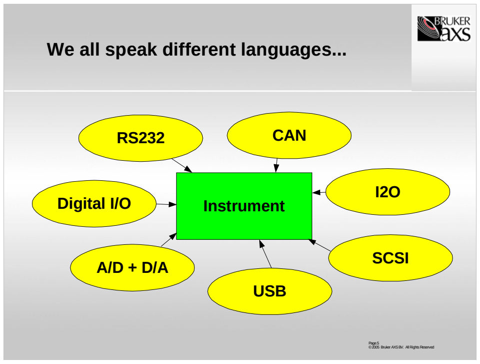
A crystallographic instrument consists of many parts that should collaborate. Having all of it integrated into one instrument is not strictly necessary, but it does allow experiments that would otherwise be absolutely impossible.

The minimum integrated system consists of a shutter, a goniostat and a detector (if the detector has an electronic shutter, the shutter can even be left out, but CCD dectectors do not have that possibility.

Integrating the cryostat will allow one to do measurements at different temperatures in an automated fashion, and also to find out afterwards at what actual temperature a measurement was performed.

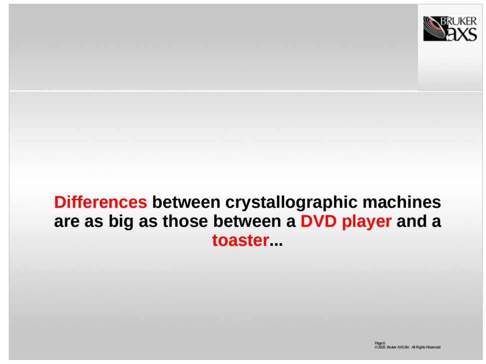
Integrating the generator will allow one to do measurements at different generator settings in an automated fashiom, and also to find out afterwards at what actual settings an experiment was performed.

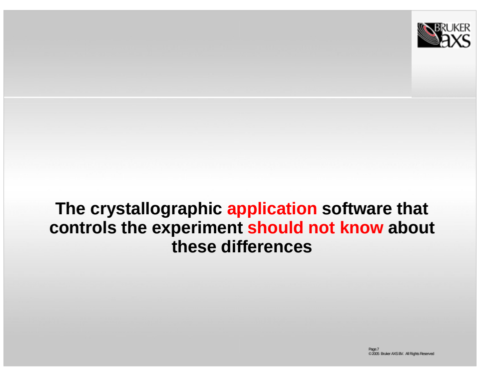
Integrating a sample robot will allow one to do measurements on a series of samples in an automated fashion, and also to find out afterwards for which sample an experiment was performed.

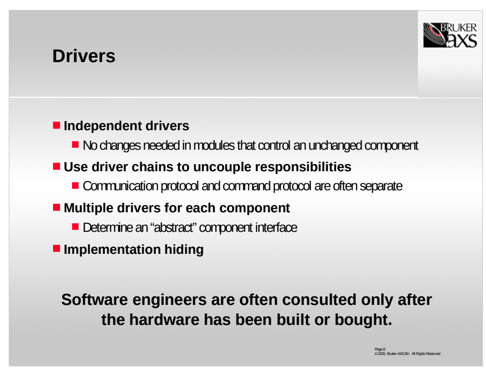


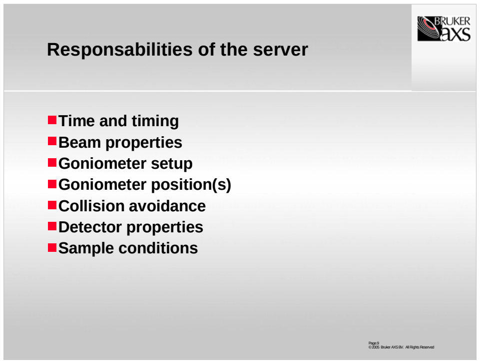
The big challenge for the integration of all of the parts into a properly configured instrument is that all the protocols differ, and different systems use other protocols. Tomorrows shutter may not use the same protocol as the one used today.

It becomes even more complicated if some components share one (RS232) connection like a generator that must be addressed via a goniostat, or a shutter that must be controlled via a generator.

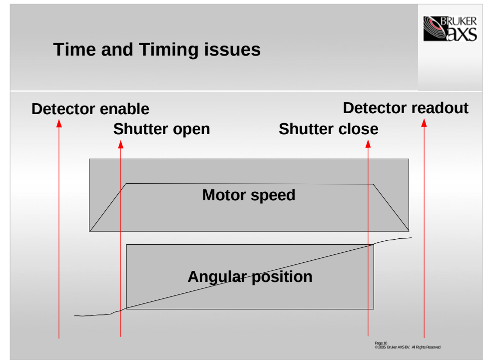








We have already listed some responsabilities of the server software. We will now go briefly into each of the listed issues.



Timing issues are critical to the accuracy of the measurement. In a typical single crystal diffraction experiment the crystal is rotated with a constant speed during the measurement. Because the motors can not accellerate infinitely fast they are actually spun back from the starting position to ramp up their speed, and when they reach the starting position at the programmed rotation speed, the shutter will be opened. When the end position of the rotation has been reached, the shutter will be closed, the motors will be ramped down, and the CCD detector will be read out.

The properties of the X-ray beam depend on the X-ray tube (RAG/Sealed, type), optic (monochromator/multilayer optic), and collimator holes.

The application program should not be concerned by the exact hardware; instead the beam must be described in the form of some reusable parameters that can be used to describe any future and past beam

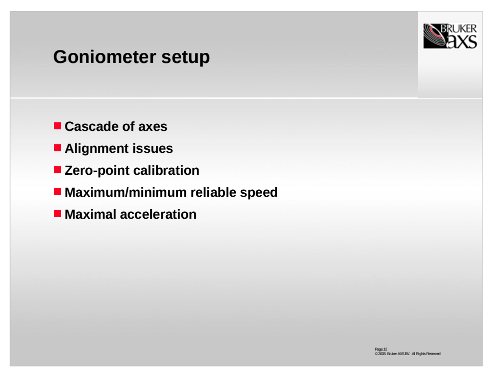
Wavelength

Polarization

Convergence/Divergence

Slits

Pinholes



Which axes, and connected how

Alignment

How fast/slow can the axes move

How accurate is each axis

Goniometer positions

- Conversions between different formalisms
- Where are you, where can you go
- Virtual goniostats: rotate around non-existing axes

If any conversion is done by the system, this may mean problems later.

Simple example: a rotation from phi=0 to phi=90, was that a positive rotation around 90 degrees? or 450 degrees? or -270 degrees?

Interpolation is a different issue:

•Kappa <-> Chi conversion

•Virtual goniostat

We assume that a rotation is linear in speed in the given axes!

Where are you, where can you go: -140 to -90 and +60 to +140 in omega, So: we can not scan from -130 to +130!

Collision avoidance

- Where can we go
- How do we get there: efficient, safe
- Margins
  - Inaccuracies in description and in the mechanics
  - High-speed scans
- Escape from a collision
- Different sample stages
- Objects in the way

The more complicated the system, the more areas there are that can not actually be reached.

To calculate where one can go and where not is a very simple calculation in principle, it just requires an accurate description of how the machine looks. The accuracy is limited by mechanical tolerances and by how much time one wants to spend calculating.

It is much more difficult to calculate an efficient route from one to another point if not all points on the intermediate straight track are accessible.

Detector properties

- Dark current
- Bad pixels
- Distortion and alignment
- Sensitivity
- Accuracy
- Point spread

Some corrections applied at the application level?

Distortion correction

- pixel coordinate system is 2-dimensional
- mm coordinate system is 3-dimensional
- Use of other coordinate systems does not make sense.
- Distortion and alignment of the detector are “inseparable”
- Transformation for the distortion uses
  - spline
  - Normalized polynomials
  - Chebychev polynomials

We had a talk about crystallographic coordinate systems, but the instrument has coordinate systems as well. The goniometer position is one, another is the detector that can be seen as an array of pixels, or as an array of 3D diffracted rays.

The transformation between these two consists of distortion correction and alignment of the detector.

Application writer's dream

- The detector should be ideal:
  - Perfectly aligned
  - No point spread
  - No distortion
  - Same sensitivity everywhere
  - No dark current
  - Counting statistics
  - Not sensitive for anything but the X-rays from the sample.

Based on the image of an imperfect detector, one can correct for all effects in order. But is that really the way to go? Some of the corrections will really deteriorate the data: e.g. correcting for image distortion will make the pixel size larger than it really is (adding noise!)

Counting statistics is lost by corrections.

Furthermore it is possible during the integration to be “intelligent” about what is happening. Correcting for an unreliable pixel inside a peak can use a different algorithm than in the background.

Also, one should make an important distinction between how the image looks to the naked eye, and the quality of the integrated data. The eye is sensitive (and extremely sensitive) only to nuances in the background, and not at all to peak height!

Sample conditioning

- Temperature
- Humidity
- Pressure
- Magnetic field
- ...

Sample conditioning may look a trivial requirement for most people, as the instrument may be operated at the same condition 99% of the time, but integration will give a proper reporting in the metadata, and is very useful in an inaccessible synchrotron system. It also allows for automation.

Not all responsibilities are easily appointed to the instrument or to the application!

We discussed a lot of the responsabilities of the server, but it is not always obvious where to place features.

An example are human interpretations of crystal looks. Another is the data collection strategy that depends on the detail of the instrumental buildup.

Conclusions

- Hardware development is as unpredictable as application requirements
- Keep a clear eye on whether the instrument or the application is responsible for an issue
- Modularizing allows for expert written code and maintainability