An SDL Framework for X-ray Spectrometer Software

Tuomas Ihme
VTT Electronics

SAM98, 29.6 - 1.7.1998

The Space2000/sw project

- A part of the technology programme Space 2000 funded by TEKES
- Goal: evaluate and adapt software methodologies for space software
- Participants: CCC Systems Oy, Patria Finavitec Oy Systems, Space Systems Finland Oy, TEKES and VTT Electronics
- Duration: November, 1996 - June, 1998
- Results:
  - An evaluation framework for software methodologies
  - An outline of component-based development of mission critical software
  - Product data management (PDM) solutions for embedded systems in space applications

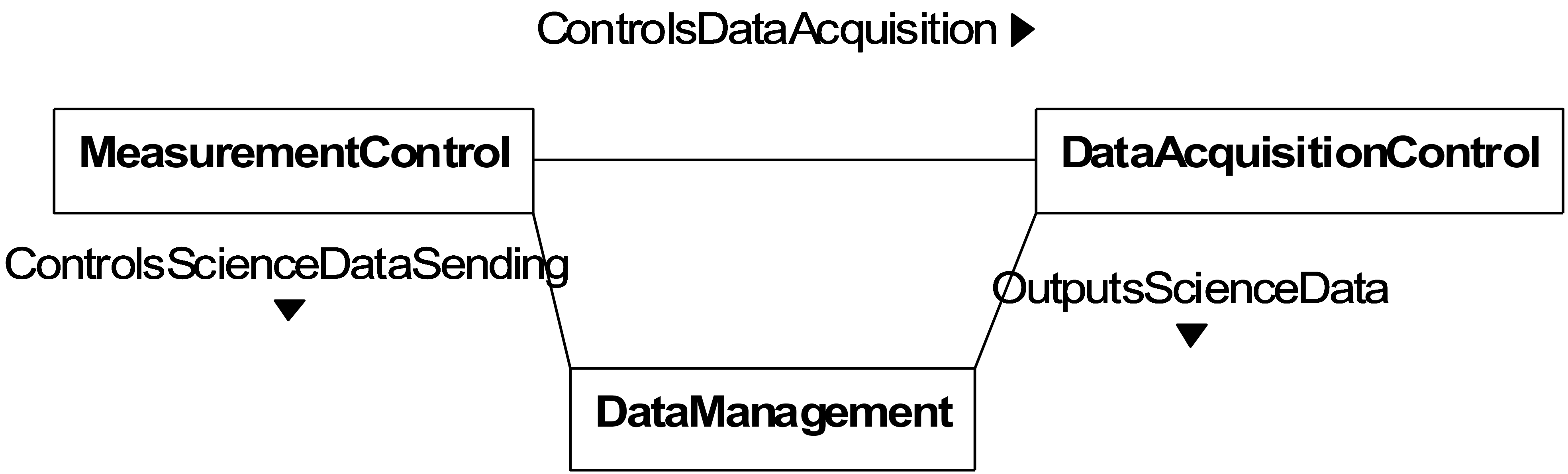
Outline

- The Measurement Control architectural pattern for designing the centralised control architecture of SDL models for spectrometer control software.
- An SDL design framework for the spectrometer controller family.
- The experience gained from the development of the SDL pattern and framework
- Conclusions

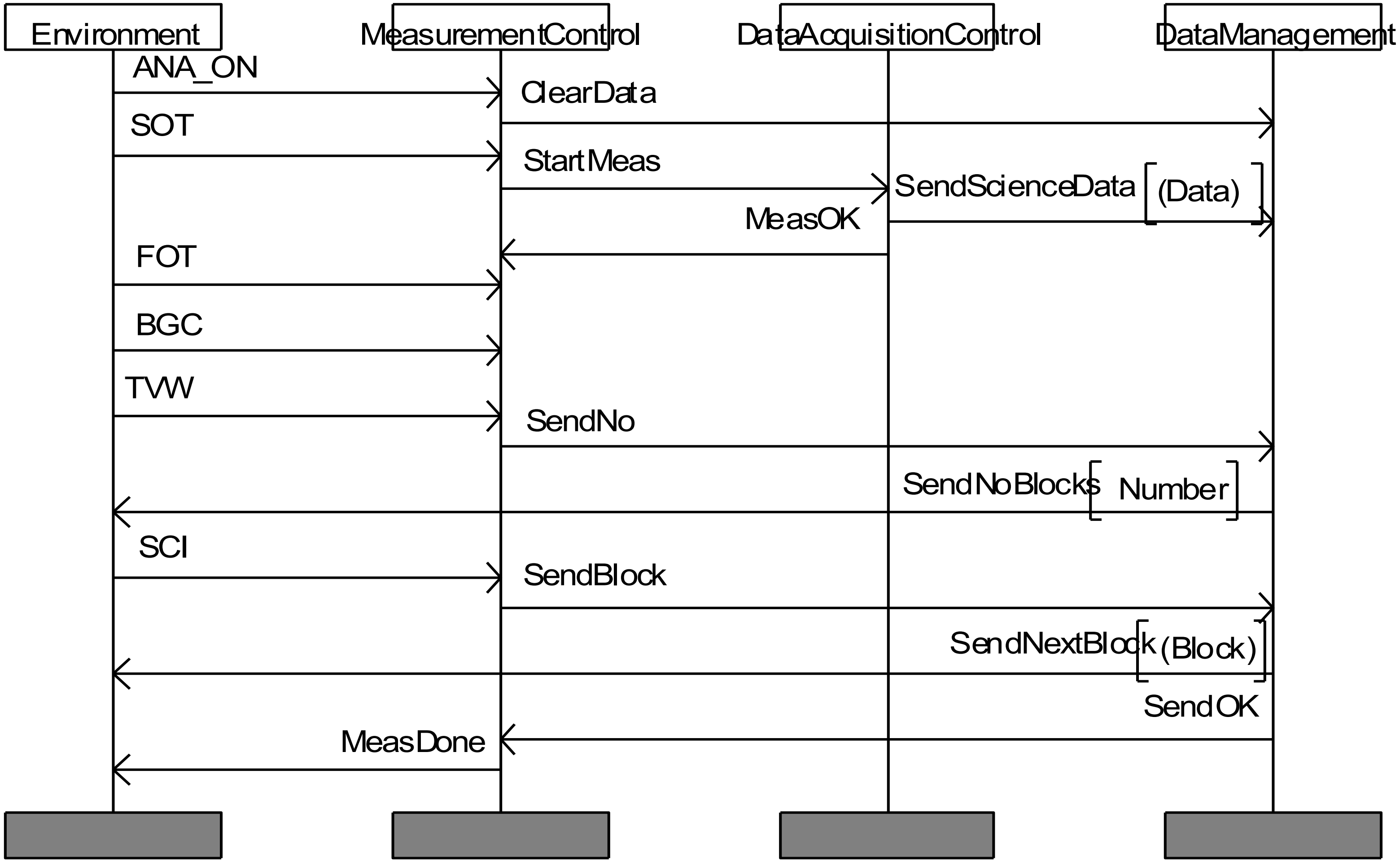
Measurement Control architectural pattern

- Intent
  - The Measurement Control architectural pattern introduces a centralised control architecture for the Measurement subsystem of X-ray spectrometer controllers.
- Motivation
  - Centralised control architecture is very common in embedded control software. The architecture is also known as master-slave architecture.
  - The complexity of control is centralised on the master. This makes it easy to modify and maintain the software.
  - The master-slave architecture is well suited to hard real-time systems requiring complete timing predictability.

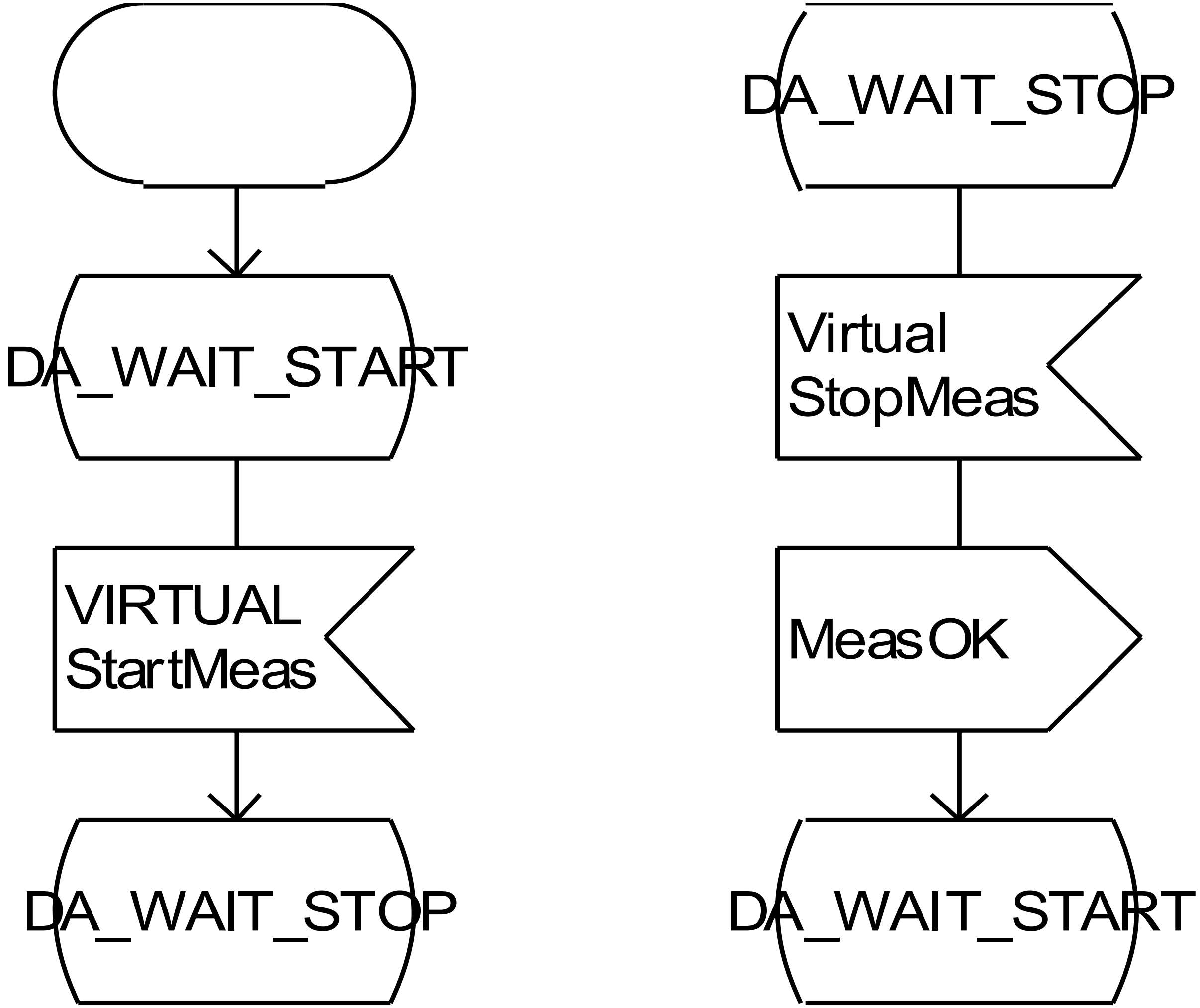
The structure of the Measurement Control architecture



A message scenario typical of the Measurement subsystem



The SDL fragment of Data Acquisition Control



Semantic properties of Data Acquisition Control

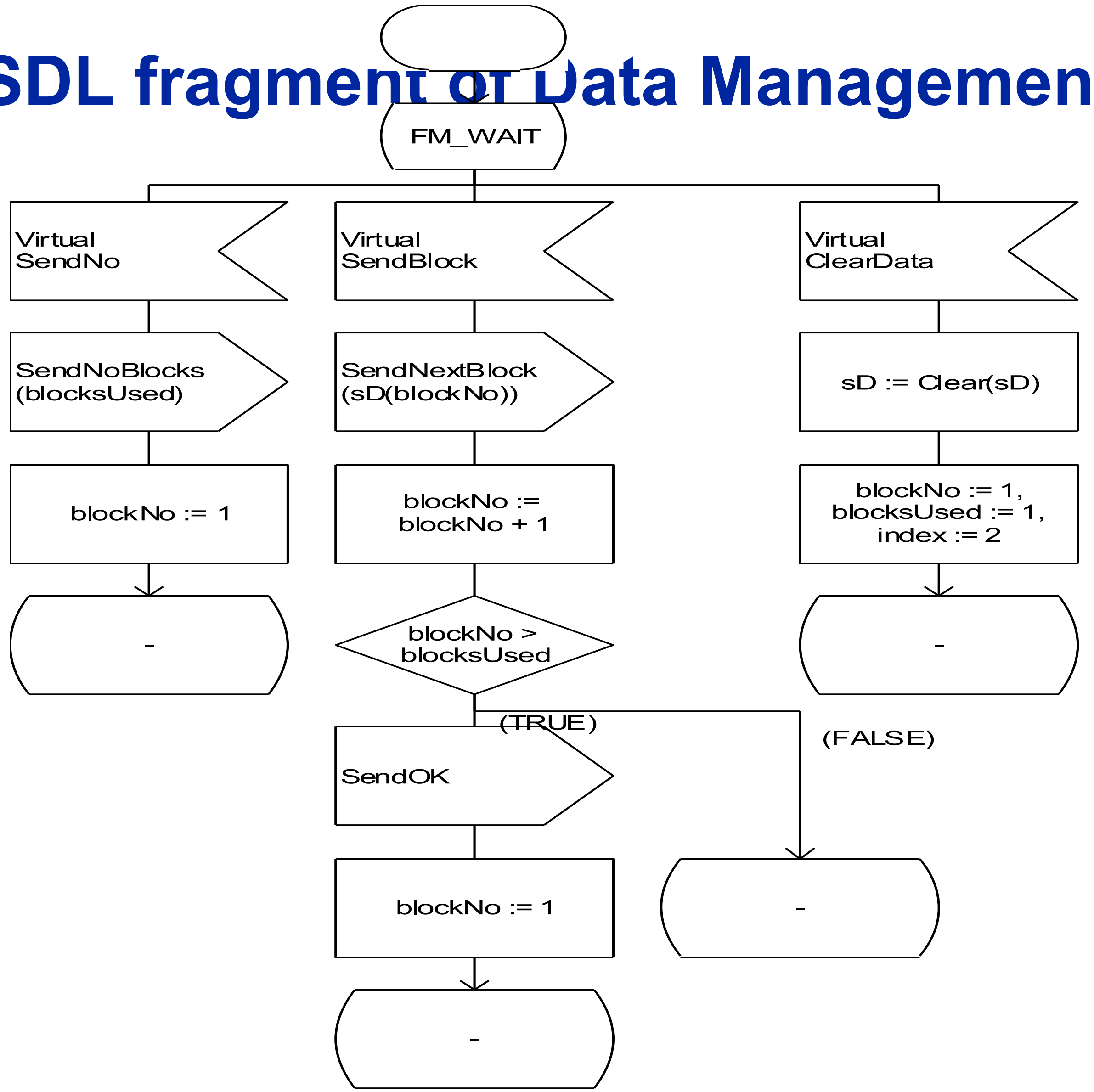
- Property 1: If the assumptions stated below hold, Measurement Control will eventually receive the MeasOK signal from Data Acquisition Control after sending the StartMeas or StopMeas signal. The assumptions are as follows:

  - The StartMeas, StopMeas and MeasOK signals are not implicitly consumed by the superclasses.
  - The transmission of the signals between Measurement Control and Data Acquisition Control is reliable.
  - The MeasOK signal will always be sent before the state DA_WAIT_START.
  - The state DA_WAIT_START will eventually always be reached.

Redefinition of Data Acquisition Control

- The embedded SDL fragment will be supplemented by additional statements for acquiring spectrum data from an array detector and sending spectrum signals to Data Management. Property 1 still holds, if the pattern is redefined by the introduction of additional statements, which do not disrupt or bypass the thread of control from predefined input to predefined output statements. However, the thread of control from the StopMeas input signal will be bypassed by threads of control that end in the MeasOK output statement and the state DA_WAIT_START.
- During observation in the DA_WAIT_STOP state the thread of control stays in the polling loop of the hardware/software interface several hours. The polling will have to be continuous in order to meet the requirements set on the data collection speed. The continuous polling will be replaced with periodic polling using a timer-triggered transition from the DA_WAIT_STOP state for the simulation purposes of the system.

The SDL fragment of Data Management



Semantic properties of Data Management

- The SendNoBlocks signal has to be sent to the environment with the number of blocks after the SendNo signal from Measurement Control. The SendNextBlock signal has to be sent to the environment with the next data block after the SendBlock signal from Measurement Control.
- Property 1: If the assumptions stated below hold, then Measurement Control will eventually receive the SendOK signal from Data Management after sending the SendBlock signal. The assumptions are as follows:
  - The ClearData, SendNo and SendBlock signals are not implicitly consumed by the superclasses.
  - The indexes and counters in Data Management are properly initialised and modified.
  - The state FM_WAIT will eventually always be reached.

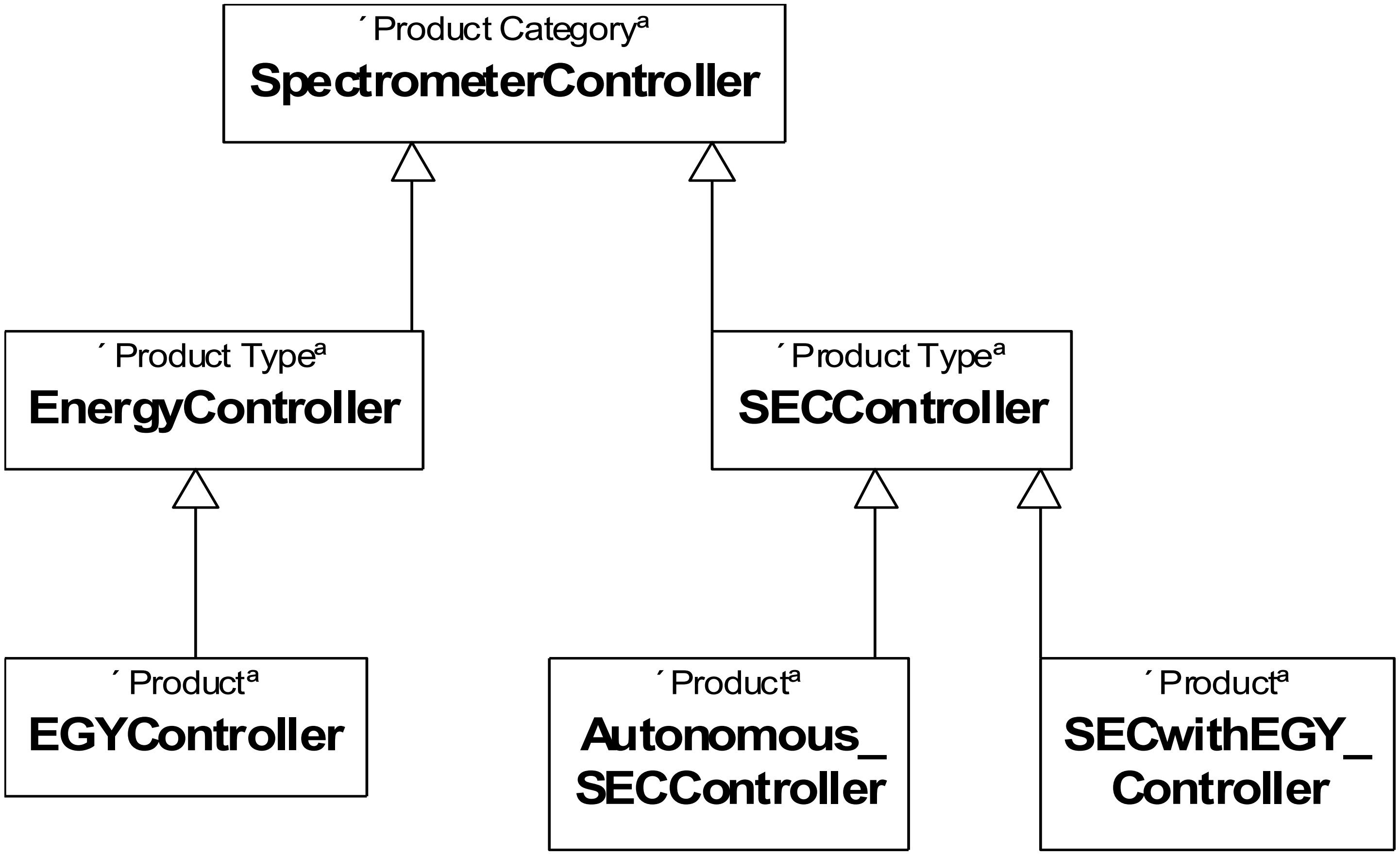
Redefinition of Data Management

- The embedded SDL fragment will be supplemented by additional transitions and statements for saving the science data sent by Data Acquisition Control. Property 2 determines the allowed redefinitions:

  - Property 2: Property 1 still holds, if the pattern is redefined by the introduction of additional transitions from the FM_WAIT state for saving the science data associated with spectrum signals from Data Acquisition Control. The indexes and counters in Data Management must be properly initialised and modified in the additional transitions. The state FM_WAIT must eventually always be reached at the end of the additional transitions.

SDL framework for spectrometer controllers

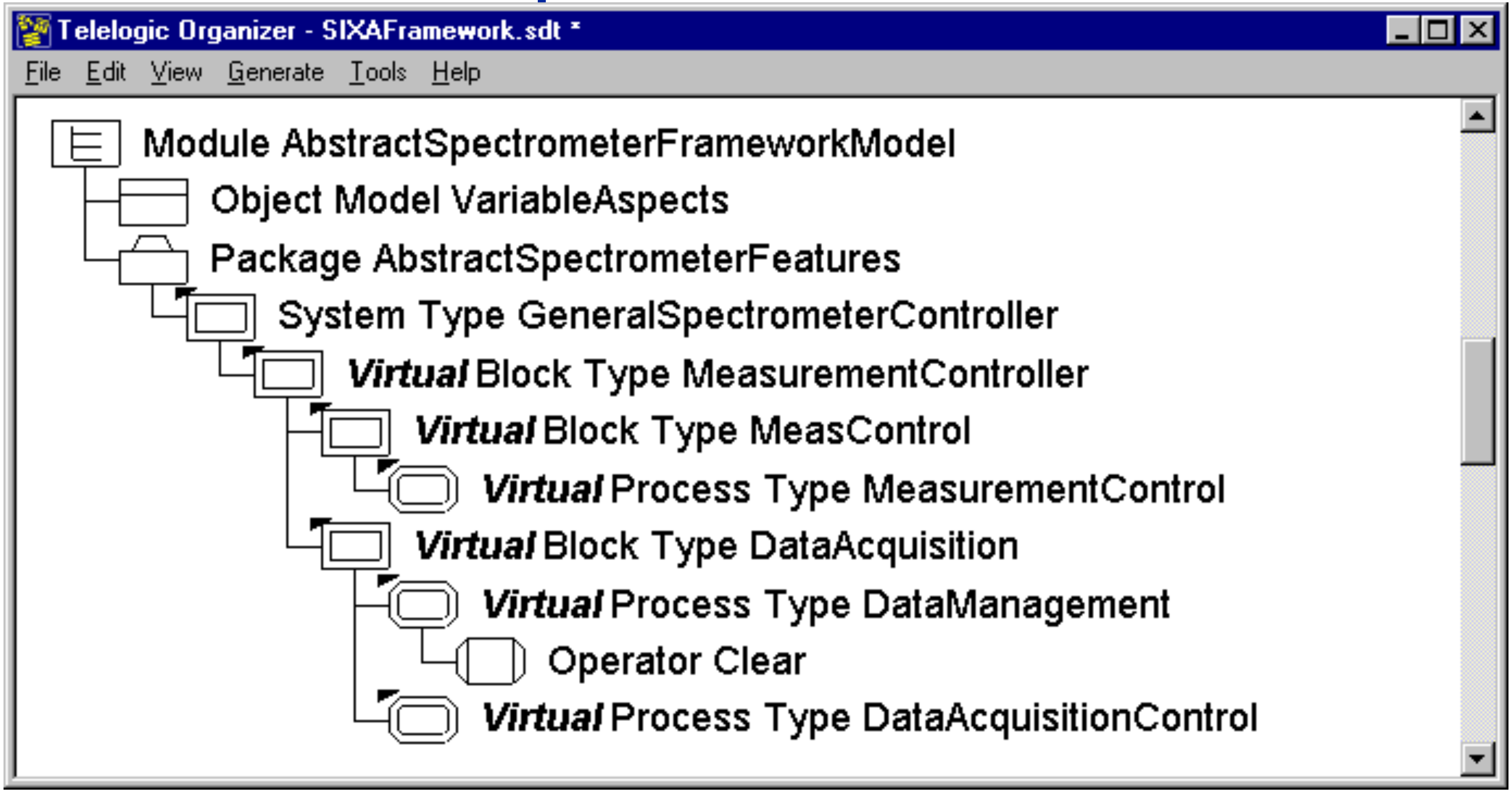
- The SDL framework includes SDL models of measurement subsystems for a family of spectrometer controllers.



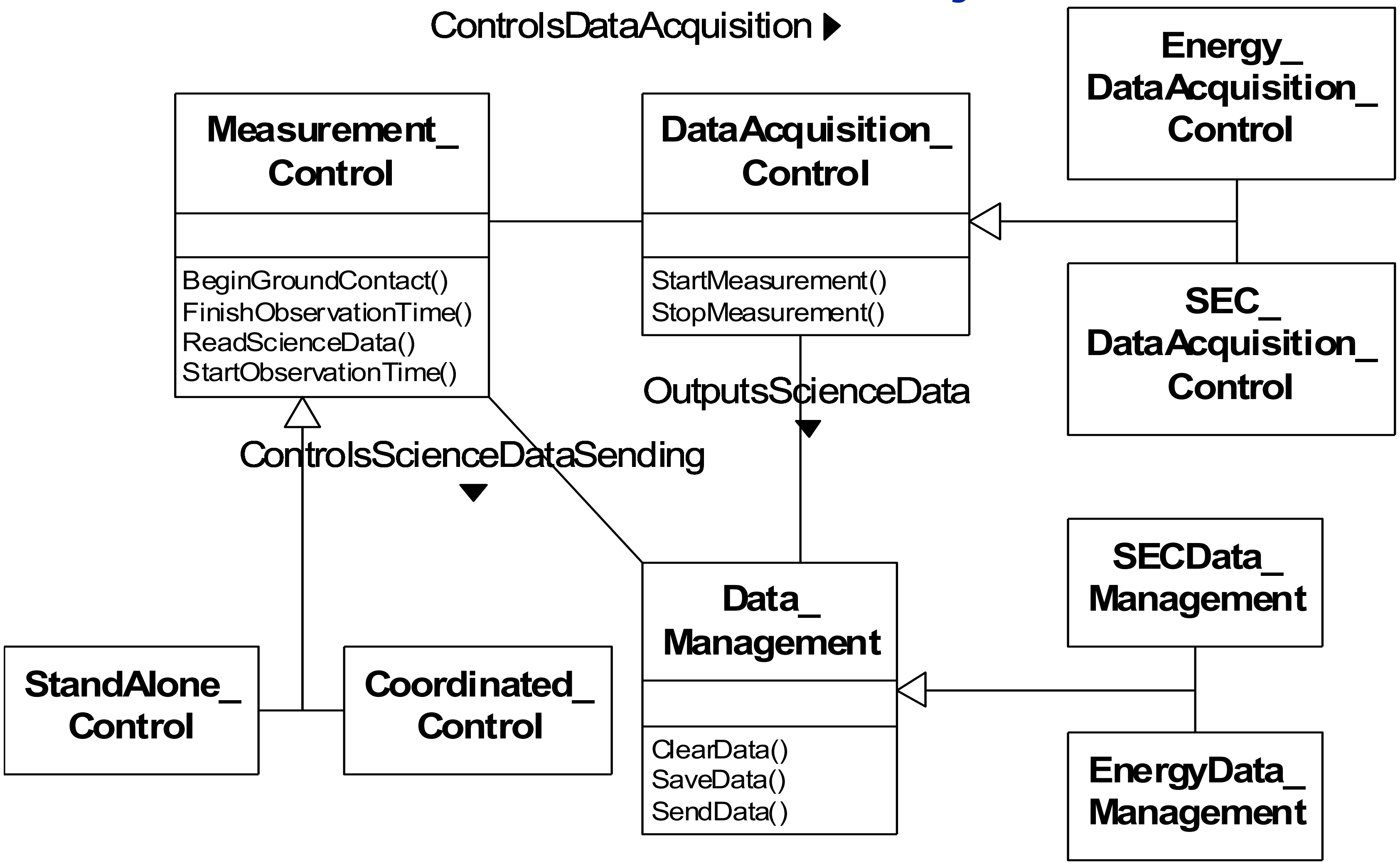
The documentation structure of the SDL framework

- The SDL models are partitioned in three modules of the Telelogic SDT tool:
  - Abstract Spectrometer Framework Model,
  - EGY Framework Definition Model and
  - SEC Framework Definition Model.

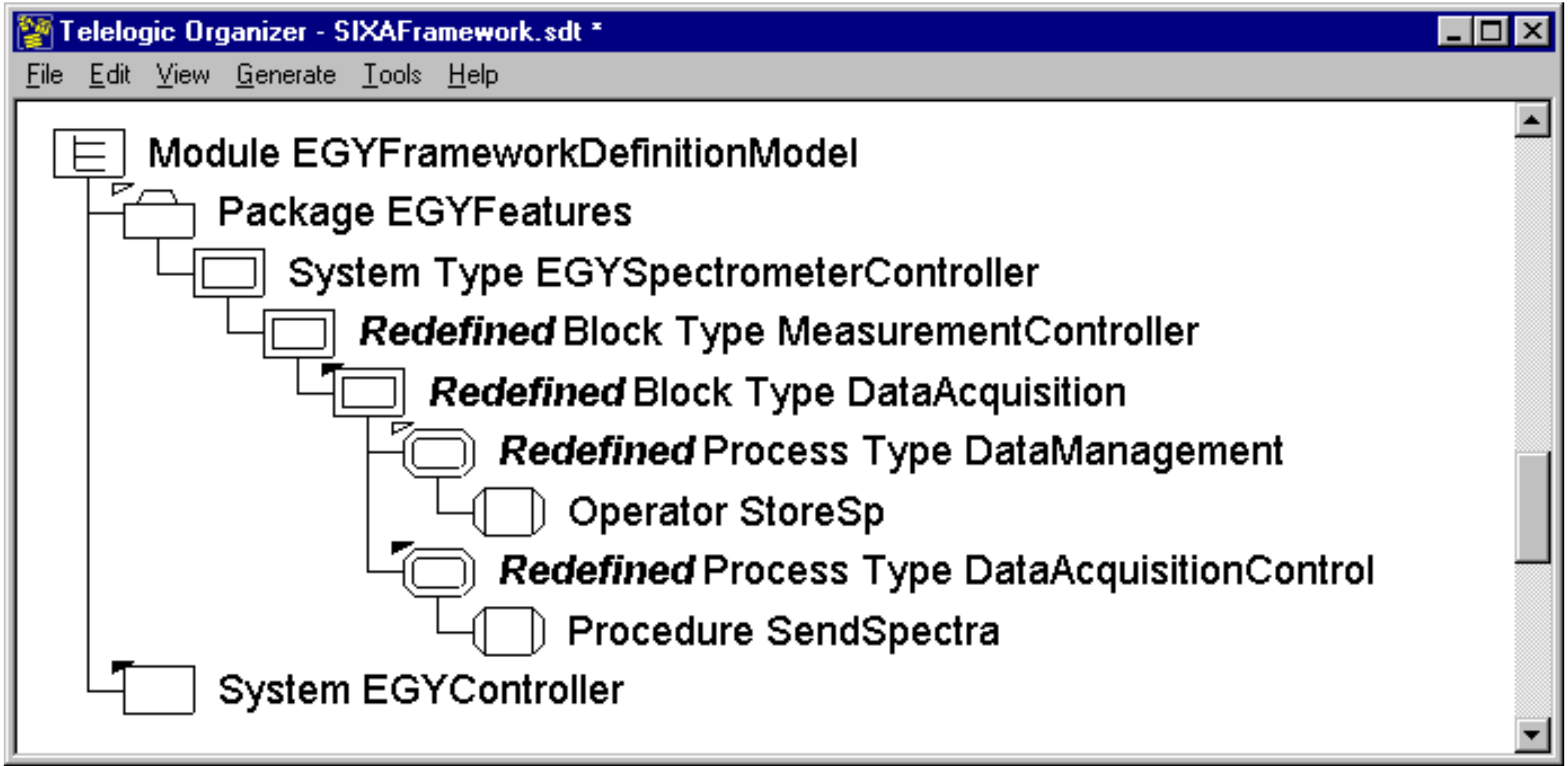
The Abstract Spectrometer Framework Model



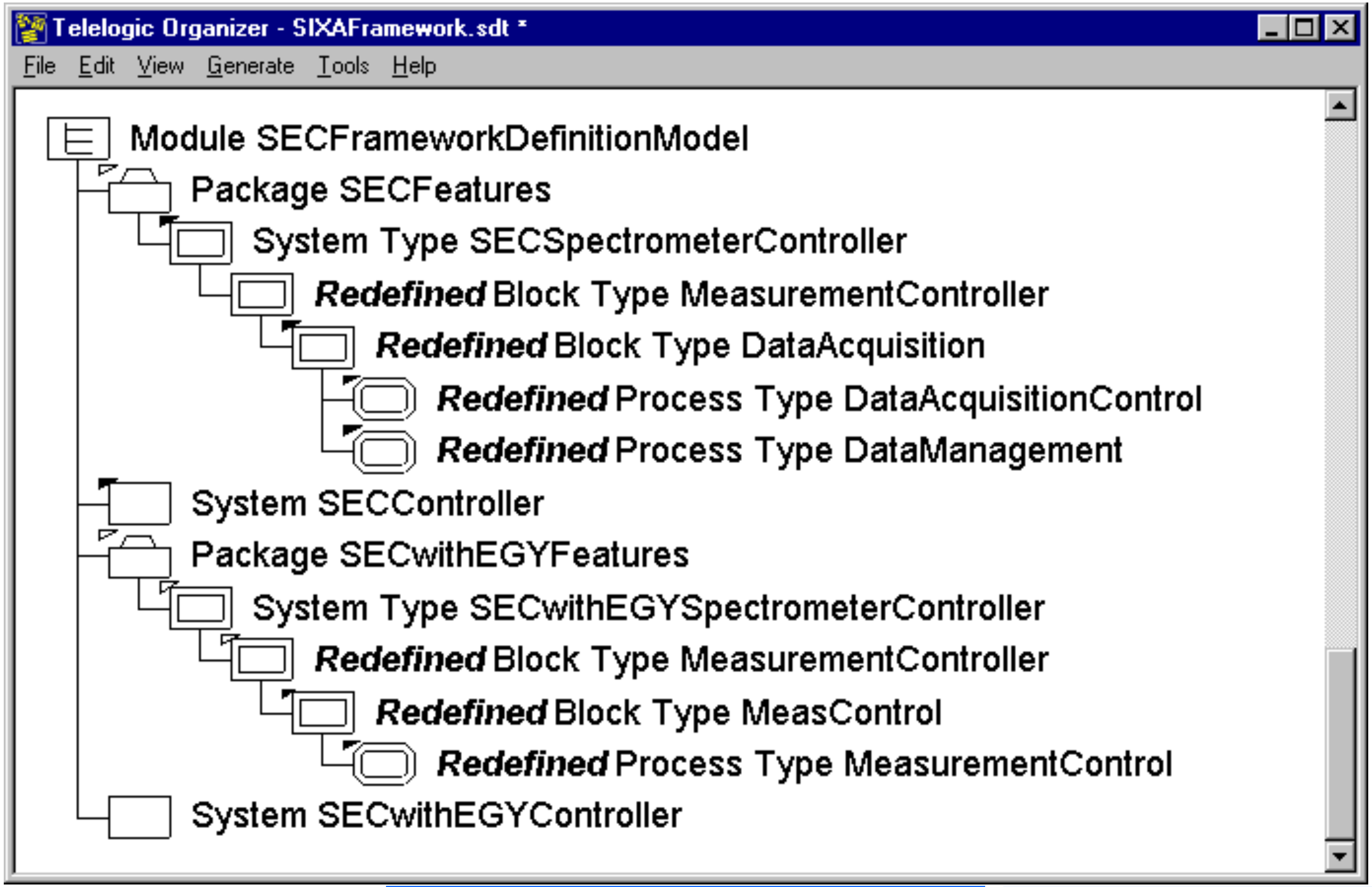
Measurement Domain Object Model



The EGY Framework Definition Model



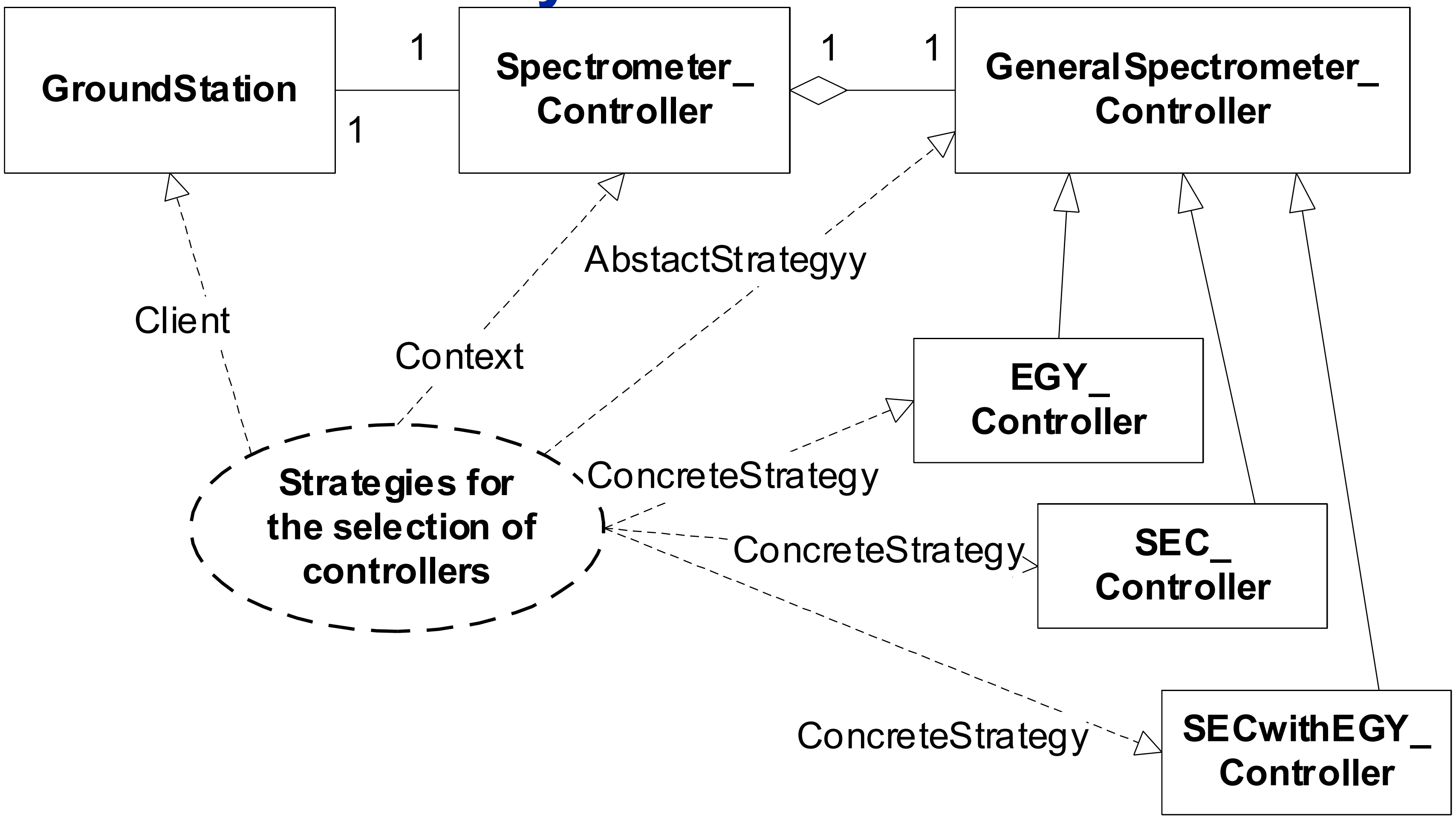
The SEC Framework Definition Model



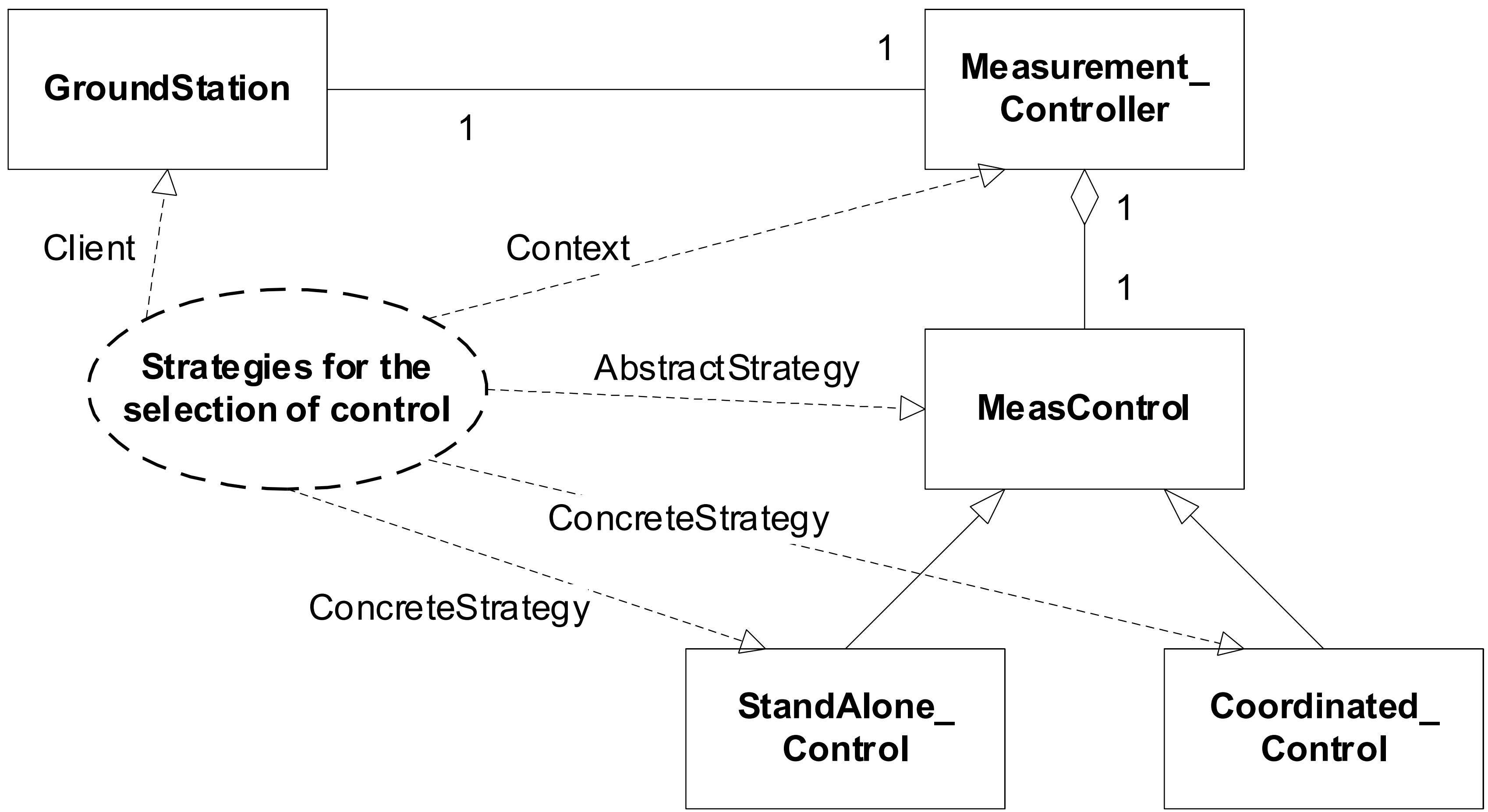
Reusability of the SDL framework

- The SDL framework supports several strategies for the reuse of SDL components.
- The dependency relationships between the component systems in the framework are well-defined and documented.
- The variable aspects of the framework are documented.

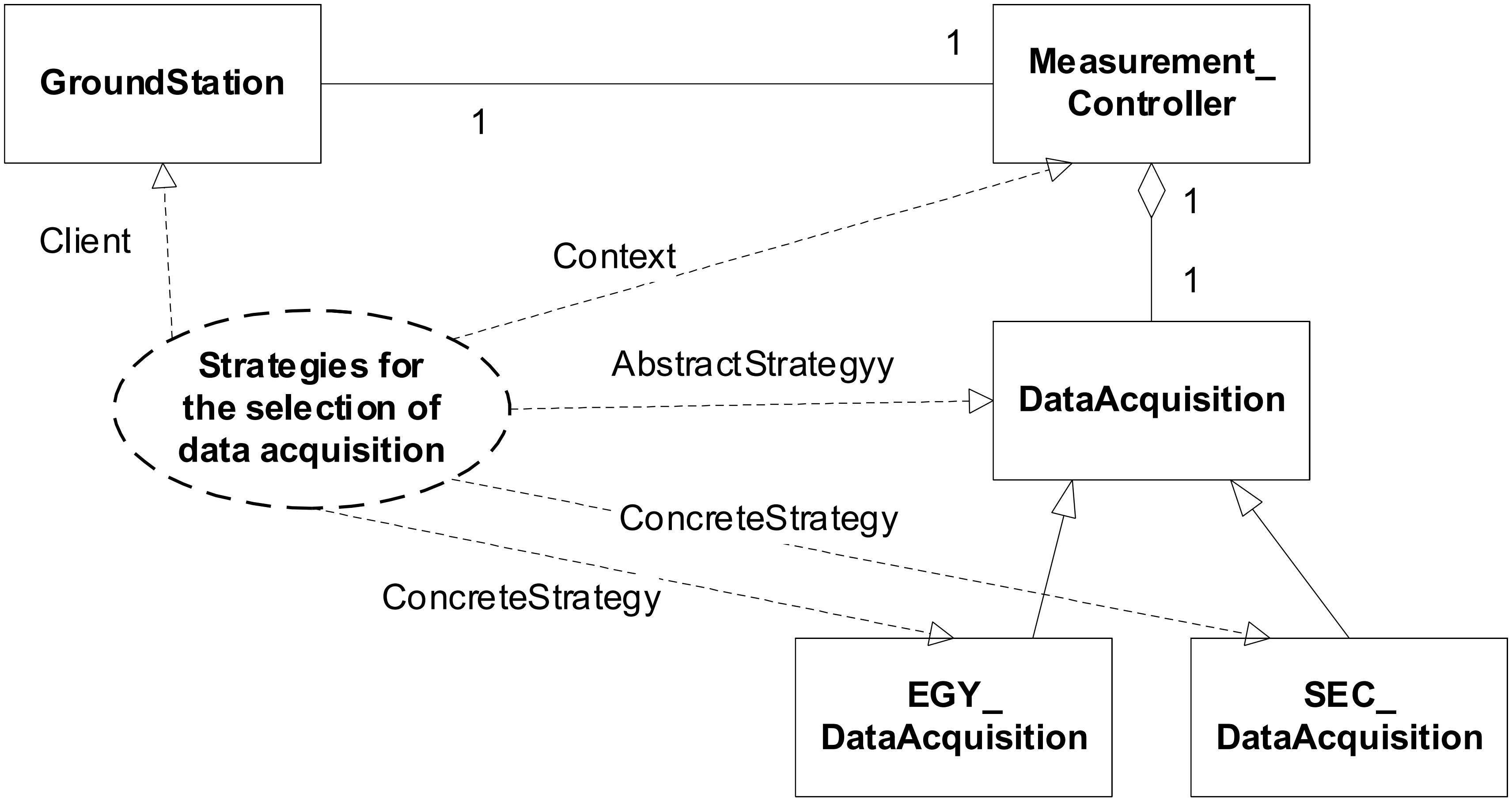
The selection of different Spectrometer Controller system models



The selection of concrete control processes

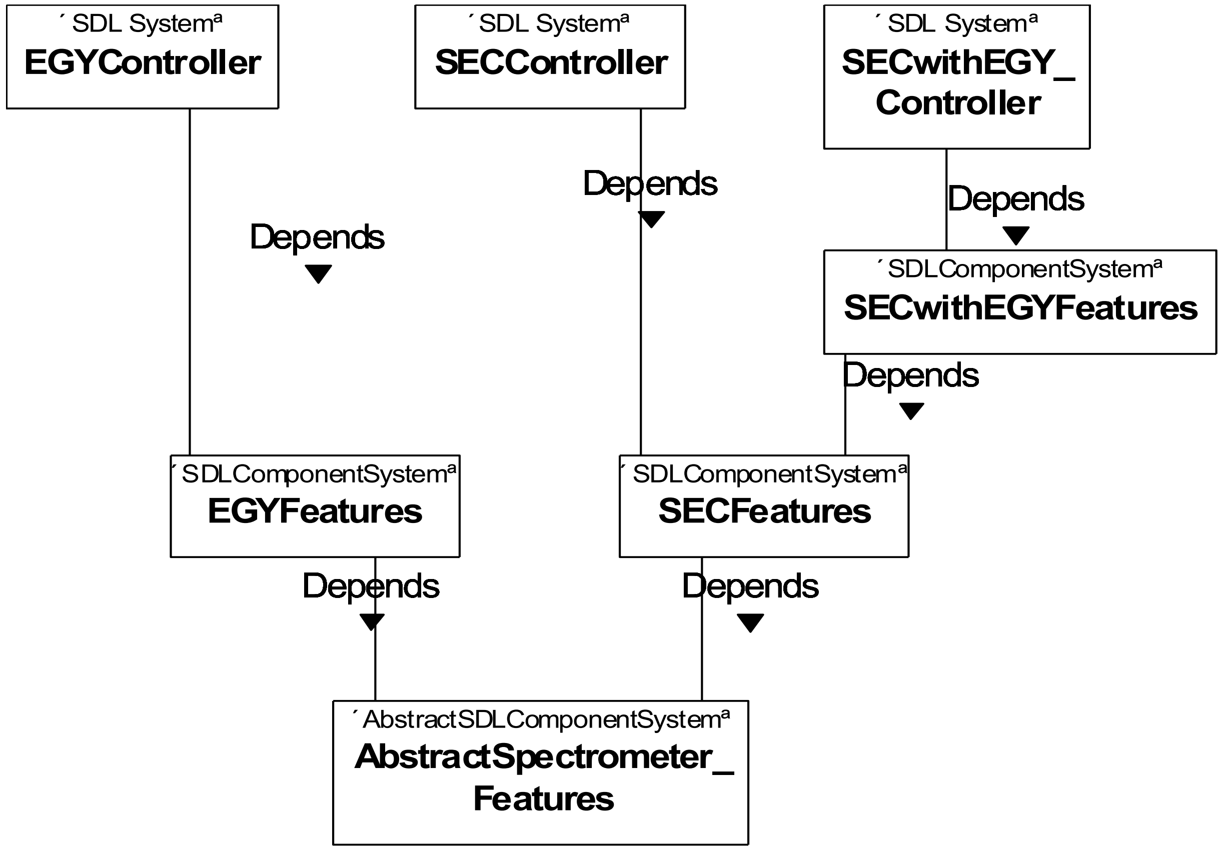


The selection of concrete data acquisition strategies



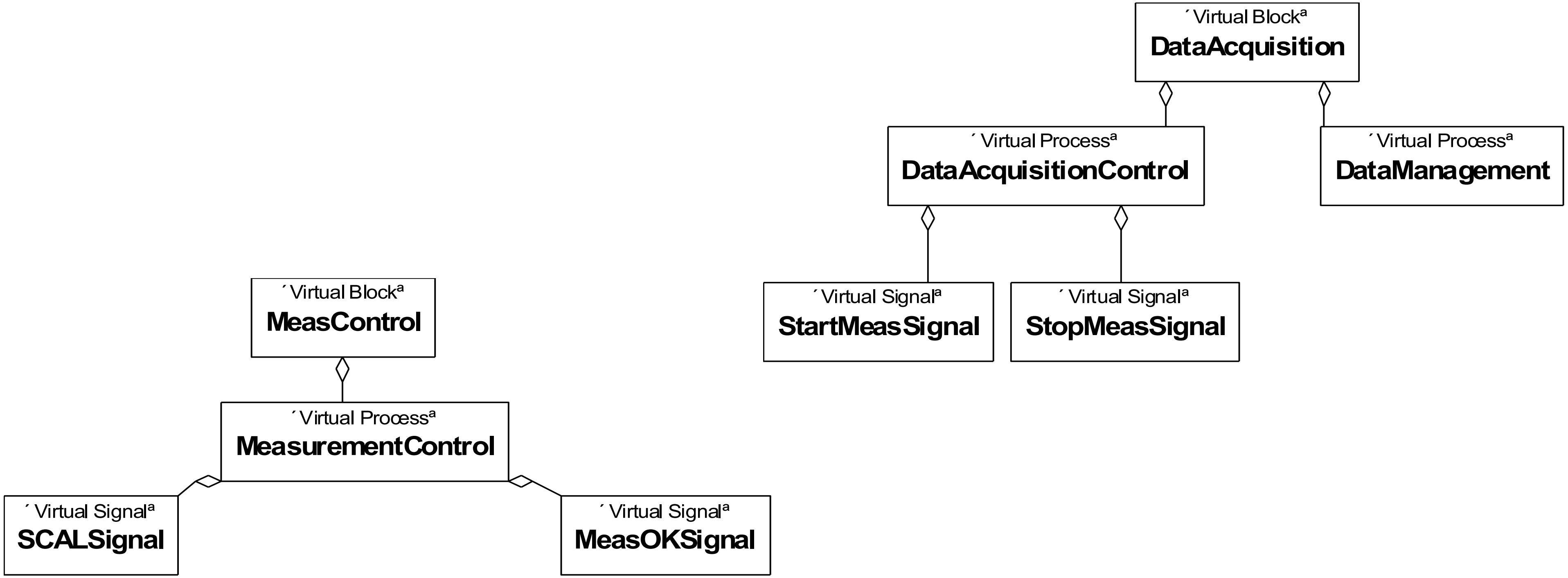
Component System Dependencies

- Dependence is highest at the top of the diagram and lowest at the bottom.



Variable Aspects

- The VariationPoint links from aspect objects to SDL components are used to identify the locations in the SDL framework at which variations will occur.



Conclusions

- The used SDL pattern approach and templates proved well suited to the rather passive slave components of the spectrometer software.
- The proposed Measurement Control pattern had an important role in designing the architecture of SDL models in the SDL framework for a family of spectrometer controllers.
- The existing strategy pattern appeared to be useful in the documenting of component interfaces and design, as well as the reuse strategies of the framework.

Conclusions (cont.)

- Problems were encountered in the description of complex master components.
  - The use of MSCs for the definition of pattern properties in addition to textual descriptions was proposed.
- Problems were encountered in the description of domain-specific architectural SDL fragments.
  - Dedicated means are needed for the documentation of variability and dependencies.
  - Special means are needed for the configuration rules of SDL components.
- Not all components of spectrometer controller software can be implemented in SDL.
  - The framework should allow attaching non-SDL components.
- The framework proved difficult to develop, understand and maintain by means of the existing CASE tools.
- Difficult to apply general design patterns in SDL design
  - Object-oriented features of SDL are specific to SDL
  - SDL lacks a sound interface support.