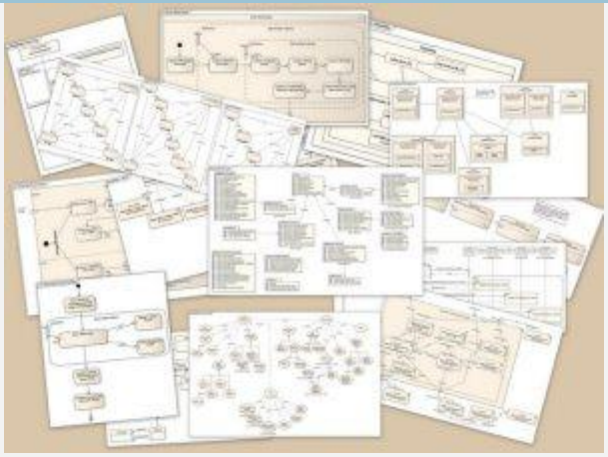
PLC Programming Approaches

Do you start with a written functional description? Do you sometimes just program by looking at the output and then work backwards till you get that output to do what it needs to do? Or do you ever use program modeling tools? Lets find out! Click here to take a one question survey and read on.

I've now read about finite state logic and tools such as StateWorks and SPLatBasic,S88 and tools like ControlDraw, Executable Unified Modeling Language, SFC, FBD, IEC 61131-3 and others, with many offering some level of modeling that can be directly implemented sometimes without writing additional code it is claimed. And yet usage is still limited.



A collage of UML diagrams

Part of the difficulty in using modeling tools is that they are not integrated into the controls programming tools, so they require translations to go from the modeling tool to the controls platform of choice. Some of the modeling tools are hardware specific running only on a PC for example.

Another difficulty is that for process jobs at least we already use the P&ID drawings or a machine layout drawing as a process model. Using another process modeling tool on top of that just adds another layer whose value is diminished.

In the industries that I've worked, it is common to write functional descriptions only when the client wants to review what we are proposing to write. Programs are typically either written using code from a similar project, often provided by the client, or by blocking it out and then coding the blocks. In addition to writing either a function description or blocking out the code requirements, I will typically write an outline of all the functions and sub functions needed and place this in the program itself as the beginning of an outline that serves as a reminder of what I'm trying to do. At the end of the project I have a series of these outlines which are left to aide subsequent readers, including myself, in understanding how the program is organized with descriptions of what each function is doing.

The blocking method of programming and general practice of writing modular code works pretty well but it can be error prone as well as being written differently by each engineer that implements such code. Also because much of the code is custom the development effort is high. The idea of using standard tools and code reuse is attractive to keep development efforts down.

In short, the level of modeling we do is limited to blocking out the code or sometimes writing a functional description at the most. At the end of the day none of this survives past the end of the project typically unless this is encapsulated in the code itself. Even then it is difficult for anyone but a programmer or technician to examine… it is difficult to share…

The need for a modeling tool that everyone can understand at a high level that allows drilling down right to actual executable PLC code is still much needed. The suggestion that I make is for those who are developing such tools to work together with the PLC manufacturers to get their product integrated in to the PLC and controller development environments. At least this is what I would like to see happen as an end user.

Source: http://hennulat.wordpress.com/2010/01/28/how-do-you-program-the-plc/