

Real-time Control Systems (RCS) Concepts

RCS is Made for Complex Systems

One of the toughest things to build is an automated system. By definition, automation is an attempt to use computers for an activity that previously has been done by humans. Human activity such as manipulating objects (grasping, lifting, transporting, sorting etc.) is relatively easy to describe in high level terms but extremely difficult to translate into computer code. Every task must be understood with incredible detail before it can be effectively programmed.

The RCS Methodology, using a unique task based approach, helps to organize domain knowledge during the initial design phase of an automated system. This initial information structure helps keep the system from becoming too complex. Using [RCS Enterprise Control Studio](#), the system design translates directly into computer code.

Human vs Robot in Space

Which is easier; teaching humans to do a task or programming a computer to do it? Remember, unlike humans, the computer has no prior knowledge of how to do anything. We must program it do to everything!

In any automated system, first the system developer must gain domain knowledge. Then this knowledge must be structured so that it does not become too complex to be understood, and then it must be programmed into the computer. No matter what programming techniques are used, there is no getting around the need to obtain very detailed system knowledge of all tasks to be performed. Usually, this is better done earlier than later.

The RCS Methodology provides the structure and techniques to accomplish this activity in a very straight forward manner. No other technique flows as naturally for the system designer from initial concept to final implementation.

RCS is Designed for Humans

The most important element in any control system is not the hardware or the software. The most important element is the human. Our ability to comprehend and organize information is the most critical and unfortunately the weakest link in any real-time control system project. RCS systems are designed in a way that is consistent with how we think, gain understanding, and retain information. It also takes into account the limits of our intelligence. The following lists the basic tenets supporting the RCS Methodology:

People tend to be **goal** oriented; it is much easier to understand tasks in terms of the intended result. In RCS systems, the means are always worked out in context with the end in mind.

Story telling is an ancient and easily understood method of conveying information. RCS Methodology lays out **scenarios** in logical sequence that effectively describes a series of tasks for automated system performance.

Unlike functional systems where coordination occurs after development, the RCS task design, which begins with repeated evaluation of scenarios, points out very quickly where coordination must occur.

Although humans are very intelligent, limits tend to be reached when trying to comprehend too many separate pieces of information. The RCS rule of thumb is **seven** +/- **two** things at the same time. Each task is kept to this optimal level of detail.

Complex systems are better understood if the various components can be related to each other. RCS's **hierarchical structure** (like a military chain-of-command) provides the best method as evidenced by its popularity in many different fields.

High level tasks are better understood if they are broken down into smaller subtasks. RCS **decomposes** larger tasks, yet each subset of a larger task maintains about the same level of complexity of any other task.

The process of abstraction allows us to take a lot of underlying detail and simplify it. RCS develops **layers of abstraction**, thus allowing us to concentrate on fewer details at any given time.

The RCS hierarchical structure, task decomposition, and resulting layers of abstraction serve to keep all system information well organized in a straight forward and intuitive way.

RCS Supports System Evolution

One of the most challenging aspects of automation is to be able to update a system after it is presumably complete. Any system, no matter it is biological or mechanical must be able to continually adapt to new (or newly discovered) circumstances. The RCS system design migrates directly into the RCS Enterprise Control Platform. Because the RCS Methodology prescribes a naturally intuitive control structure, it is very easy for the original developer or a different person who was not involved in the initial design to continually adapt the system as needs change.

Complex systems are never fully understood or completely controlled.

No matter if the system is biological, chemical, mechanical, or electronic, the complexities of all the possible interactions of its various parts will always reveal themselves gradually, and only through repeated interactions. RCS designs are fundamentally extendable; it is assumed that every system must be easily modified without fear of system failure due to unexpected side effects.

Our world model never matches the real world

As soon as we think we understand something, reality proves us wrong. The development of real-time control systems differs from other software development because of the intense interaction with real things, sensors, actuators, motors, etc. The system is connected and interacts with physical objects. This means that we cannot long maintain any illusions if our system design is flawed. RCS systems offer the clearest possible picture of the control design, so that as our understanding becomes more correct, it is easily added to the control structure.