Chapter 2
Requirements

INTRODUCTION

Model based controls system development starts from product requirements. The highest level and most abstract product requirements originate from the customer needs and the marketing analysis. The organized and analyzed customer needs are translated into the technical system requirements through applying functional analysis to start the product development: what functions the product needs to provide to meet the customer and market needs. The customer and market needs to be satisfied by the product are chosen on the other side based on the company's resource, product development cycle, product line development plan, and marketing strategies.

After the system level requirements are decided, the control subsystem level requirements need to be defined to begin the control system development. For embedded control systems, which is focused in this book, the overall control functions are implemented in software running on a hardware computing platform within the product, the controls software, as part of the overall product application software which also includes other software modules such as graphical user interface, and network communications, etc. So control subsystem requirements are in effect the requirements on the controls software.

Requirement flow down first starts from the system level to the subsystem level. From the product structure perspective, the controls subsystem is comprised of all elements (hardware, software and firmware) involved in providing and supporting the dynamic behaviors and functions of a product

DOI: 10.4018/978-1-5225-2303-1.ch002

defined by the system level requirements. For example for a vapor compression cycle air conditioning system, the control subsystem includes the compressor, evaporator, condenser, indoor and outdoor fans, measurement sensors such as pressure and temperature sensors, controller board and the embedded controls software. In general, the control system hardware includes the physical system under control (called the plant in controls term), actuation and sensing devices. In the air conditioning example, the plant is the assembly of the compressor, evaporator, condenser and pipes connecting them; assume the evaporator is equipped with a thermal expansion valve for superheat control at the compressor inlet and the compressor is controlled by a variable frequency speed drive, then the compressor variable frequency speed drive is the actuator which regulates the compressor speed; if fan speeds are adjustable instead of being fixed, then the various fan drives are actuators for changing the fan speeds too; pressure and temperature sensors installed at various locations in the system are the sensing devices for measuring controlled variables, for example the supply air temperature, and for monitoring internal variables for safe operation, for example the compressor suction superheat to prevent compressor from flooding. Based on the product structure design, the control subsystem components are identified along with its dynamic behaviors and interdependencies. The control subsystem requirements are collected, derived and then separated from the product system requirements, which describe the controlled system behaviors and performance for intended use cases. At this point, the interfaces between the control software and the rest of the application software needs to be thoughtfully defined based on the overall software application structure, for example the way in which the data is managed and shared among different modules of the application software.

The requirement flow down from system to control subsystem is further followed by the flow down of the requirements from control subsystem to control algorithm for an individual control function. At this level, control subsystem component behavior and performance are defined and quantified, for example how the compressor shall be controlled and how the evaporator fan shall be controlled to provide a specific cooling function with certain performance requirements. The decomposition of subsystem level controls into component level controls is supported by the control architecture design, and is focused around functions that the control subsystem as a whole needs to provide. For example, there could be different modes for cooling, such as mechanical cooling, free cooling (using outdoor cold air), and mixed

Related Content

Challenges and Trends of Agile

Fayez Salmaand Jorge Marx Gómez (2022). *Research Anthology on Agile Software, Software Development, and Testing (pp. 1-16).*

www.irma-international.org/chapter/challenges-and-trends-of-agile/294455

Queuing Theory Contributions and Applications in Health Service: A Study in the Field of Management Problems

Alexandre Beraldi Santos, Ana Carolina Sanches Zeferino, Ilma Rodrigues de Souza Fausto, Sandra Maria do Amaral Chavesand Saulo Cabral Bourguignon (2023). *Cases on Lean Thinking Applications in Unconventional Systems (pp. 163-183).*

www.irma-international.org/chapter/queuing-theory-contributions-and-applications-in-health-service/313654

Software Defects Prediction Model with Self Improved Optimization

Shantappa G. Gollagi, Jeneetha Jebanazer Jand Sridevi Sakhamuri (2022). *International Journal of Software Innovation (pp. 1-21).*

www.irma-international.org/article/software-defects-prediction-model-with-self-improved-optimization/309735

Model-Based Control Design

(2017). *Model-Based Design for Effective Control System Development (pp. 1-31).*

www.irma-international.org/chapter/model-based-control-design/179497

Governing the Service-Driven Environment: Tools and Techniques

Leo Shuster (2013). *Service-Driven Approaches to Architecture and Enterprise Integration (pp. 210-240).*

www.irma-international.org/chapter/governing-service-driven-environment/77951