Currently, the design of distributed embedded real-time systems is growing in complexity. Moreover, embedded systems include hardware and software components, which are usually designed concurrently using distinct models, tools, specifications and implementation languages. These systems have important domain specific requirements, such as time constraints, which do not represent by themselves the expected functionalities. However, non-functional requirements affect not only the system's way to perform its functionalities, but also the overall design success. It is difficult to deal with them during the whole design because usually a single non-functional requirement affects several distinct components. This thesis proposes an automated integration of design phase of distributed embedded real-time systems, focusing on automation systems. The proposed approach uses Model-Driven Engineering (MDE) techniques along with Aspect-Oriented Development (AOD) and previously developed (or third party) hardware and software platforms to design distributed embedded real-time systems. AOD concepts allow a separate handling of functional requirement from non-functional ones, improving the produced artifacts modularization (e.g. specification models, source code, etc.). In addition, this thesis proposes a tool to support an automatic transition from initial specification phases to the following implementation phases. This tool uses a set of mapping rules, describing how elements at higher abstraction levels are mapped (or transformed) into lower abstraction level elements. The obtained results show this work's effectiveness to improve the design of distributed embedded real-time systems.