Abstract

„Software is everywhere“. This fact applies almost every area of our daily life. Software can be found within cell phones, refrigerators, coffee machines, toasters, airplanes or automobiles. Very often, the user of these systems does not even know about the presence of software. Within the anti-skid system for example, the driver usually does not know, that software coordinates the braking procedure. A specific sensor measures the speed of the wheels and sends these data to an electronic control unit (ECU). The software within this ECU will analyze the data and will send some control values to the brake cylinder, if the wheels are blocking. Based on these data the brake cylinder will be released and the car will not break out. This system is called a *mechatronic system* and combines elements from the information engineering, electrical engineering and mechanical engineering discipline.

The development of these mechatronic systems is a very big challenge for the engineers. All advantages, disadvantages and interactions of the different disciplines have to be considered, in order to provide an optimal functionality. To reach this goal, the integration of these disciplines has to be done very early within the development process.

While the system design several different modelling languages are used to describe different aspects like requirements or structure and behaviour of the used elements. Some of these models are build upon others, some are specified at the same time. Two of these modelling languages are the *function hierarchy* and the *system structure*. The function hierarchy is used to describe the functional requirements of the mechatronic system and the system structure is used the model the elements which will fulfil the functional requirements.

The analysis of these two modelling languages shows, that a lot of interaction exists. That means that changes in one of the models will invoke changes in the other one. For example, if a functional requirement (in the function hierarchy) will be deleted, the appropriate system element has to be deleted too (in the system structure). This way of consistency is usually done by the engineer. There is no formalism to do this automatically.

Further there is no support in order to select the appropriate system elements, which should be used to fulfill the functional requirements. Usually the engineer selects the system elements based on his knowledge and experience. This becomes more and more difficult, because of the interdisciplinary nature of mechatronic systems.

A solution for these problems is presented within this thesis. Therefore three emphases were defined. First, the formalization of the function hierarchy and the system structure. Second, the definition of a search algorithm to locate the best system element for a given functional requirement. Third, the definition of an algorithm to guarantee the consistency between the two models.