Abstract

This dissertation presents the design and implementation of a new block diagram programming language, Bloqqi, for building control systems with focus on variability. The language has been developed in collaboration with industry with the goal of reducing engineering time and improving reuse of functionality.

When building a control system for a plant, there are typically different variants of the same base functionality. A plant may have several variants of a tank, for example, one variant with heating and another one without. This dissertation presents novel language mechanisms for describing this kind of variability, based on diagram inheritance. For instance, Bloqqi supports specifying what features, like heating, the base functionality can have. These specifications are then used to automatically derive smart-editing support in the form of a feature-based wizard. In this wizard, the user can select what features the base functionality should have, and code is generated corresponding to these features. The new language mechanisms allow feature-based libraries to be created and extended in a modular way.

This dissertation presents techniques for implementing rich graphical editors with smart editing support based on semantic analysis. A prototype compiler and graphical editor have been implemented for the language, using the semantic formalism reference attribute grammars (RAGs). RAGs allow tools to share the semantic specifications, which makes it possible to modularly extend the compiler with support for advanced semantic feedback to the user of the graphical editor.