ENHANCEMENTS OF STATECHART-MODELING—
THE KIEL ENVIRONMENT

Steffen Prochnow and Reinhard von Hanxleden
{spr, rvh}@informatik.uni-kiel.de
Dept. of Computer Science, Christian-Albrechts-Universität Kiel
http://www.informatik.uni-kiel.de/rtsys/

Abstract

The Kiel Integrated Environment for Layout (KIEL) is a prototypical modeling tool to explore novel editing, browsing and simulation paradigms in the design of complex reactive systems.

1. Introduction

Modeling systems based on semi-formal graphical formalisms, such as Statecharts [5], has become standard practice in the design of reactive embedded devices. However, the modeling of realistic applications often results in very large and unmanageable graphics, severely compromising their readability and practical use. To overcome this, we present a methodology to support the easy development and understanding of complex Statecharts.

2. The KIEL Environment

The Kiel Integrated Environment for Layout (KIEL) environment [11] is a prototypical modeling tool to explore novel editing, browsing and simulation paradigms in the design of complex reactive systems. KIEL is not restricted to a specific Statechart dialect; so far, it has been adapted to SyncCharts/Safe State Machines (SSMs) [1], Stateflow [12] and UML-Statecharts [6].

Statechart Layout: A central enabling capability of KIEL is the automatic layout of Statecharts, which computes bottom-up layouts at each hierarchy level using GraphViz [4]. This transforms any given Statechart to a standardized layout (Statechart Normal Form, SNF) that is compact and makes systematic use of secondary notations to aid readability.

Statechart Editing: As an alternative to the classic, low-level WYSIWYG graphical editing paradigm, KIEL provides a structure-based graphical editor, which applies high-level editing commands (e.g., “add successor state”) to

![Diagram of KIEL Environment](image)

Figure 1: Different Representations of an SUD Example [1]
Figure 2: Screenshot of KIEL as it simulates an SSM

KIEL’s provides novel editing, browsing and simulation paradigms to enhance the comprehension of the system under development. The feedback we have obtained so far regarding the concepts of editing, the SNFs, and dynamic Statecharts has been quite positive. This has also been supported by an empiric study [9].

Regarding ongoing and future work, there are numerous ways in which to extend the capabilities of KIEL. Currently, we are focussing on the extension to data-flow diagrams, similar to SCADE or Simulink; we are also considering how to integrate the KIEL capabilities into generic modeling frameworks, such as the Eclipse IDE.

References