The consistent specification of reactive, context-aware systems is still a challenging and error-prone task. One reason for this observation is, that those systems are typically involved in complex user interactions and support multi-variant work flows of humans and other technical systems. On the one hand, it is necessary to capture all system functions required for each supported activity of the work flow. On the other hand, interrelations between system functions such as their order of execution or mutual influences have to be considered as well. The idea of modeling data and control flows occurring within a software-intensive system and its environment by means of a common and precise graphical notation has been present for a couple of years now. Most modeling techniques in practice like, e.g., UML's Activity Diagrams merely offer an 'appealing' graphical syntax without including a precise mathematical interpretation for the behavioral modules -- denoted processes in the following -- exhibiting these data and control flows. This causes ambiguous model interpretations, which cannot be readily resolved. On the other hand, formally founded description techniques like, e.g., Petri Nets can not express aspects like communication, which are relevant for the faithful description of the processes of the sorts arising in computer science. Methodically relevant concepts such as hierarchy and refinement are often not supported by such description techniques. The presented modeling approach incorporates the advantages of
both worlds: a graphical representation supporting a modular, hierarchical description in terms of processes, and a formal semantics accurately reflecting the execution of these processes -- thus laying the foundation for automatic verification and tool support. Furthermore, we sketch an elementary development approach to the model-based design of system behavior, which uses different views to structure its design and analysis. It starts with a formal, structured description of a user's work flows to be supported by the system. This structure is exploited for the construction of two complementary views: one that characterizes the relevant usage conditions of the user's work flow in terms of context, and another view describing the required system behavior in that context.

Stichworte: Work Flow; Context; Denotational Semantics; Composition; Laparoscopic Cholecystectomy

Jahr: 2009

Jahr / Monat: 2009-06-19 00:00:00

Seiten: 37

Occurences:
- Einrichtungen > Fakultäten > Fakultät für Informatik > Technische Berichte > 2009

Entries: