



# Component-Based Design of Embedded Systems

Hermann Kopetz

## ► To cite this version:

Hermann Kopetz. Component-Based Design of Embedded Systems. 8th IFIP WG 10.2 International Workshop on Software Technologies for Embedded and Ubiquitous Systems (SEUS), Oct 2010, Waidhofen/Ybbs, Austria. pp.1, 10.1007/978-3-642-16256-5\_1 . hal-01061218

**HAL Id: hal-01061218**

**<https://inria.hal.science/hal-01061218>**

Submitted on 5 Sep 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Component-Based Design of Embedded Systems

Hermann Kopetz

TU Wien, Austria  
hk@vmars.tuwien.ac.at

**Abstract.** In many engineering disciplines, large systems are built from prefabricated components with known and validated properties. Components are connected via stable, understandable, and standardized interfaces. The system engineer has knowledge about the global properties of the components-as they relate to the system functions-and of the detailed specification of the component interfaces. Knowledge about the internal design and implementation of the components is neither needed, nor available in many cases. A prerequisite for such a constructive approach to system building is that the validated properties of the components are not affected by the system integration. This composability requirement is an important constraint for the selection of a platform for the component-based design of large distributed embedded systems.

Component-based design is a meet-in-the middle design method. On one side the functional and temporal requirements on the components are derived top-down from the desired application functions. On the other side, the functional and temporal capabilities of the components are contained in the specifications of the available components (bottom up). During the design process a proper match between component requirements and component capabilities must be established. If there is no component available that meets the requirements, a new component must be developed.

A prerequisite of any component-based design is a crystal clear component concept that supports the precise specification of the services that are delivered and acquired across the component interfaces. In real-time systems, where the temporal properties of component services are as important as the value properties, the proper notion of a component is a hardware software unit.

In the first part of this presentation the different interfaces of an embedded system component will be introduced. The services of the component are provided by the linking interface (LIF) that must be precisely specified in the domains of value and time. The technology independent interface TII is used to parameterize a component to the given application environment. While a component provider must assure that the component works properly in all configurations that are covered by the parameter space, the component user is primarily interested in the correct operation of the concrete component configuration. The following parts will be concerned with the composition of components and the notion of emerging properties in systems of components.